

# ML meets ML<sub>n</sub>: Machine learning in ligand promoted homogeneous catalysis

Jonathan D. Hirst<sup>a,\*</sup>, Samuel Boobier<sup>a</sup>, Jennifer Coughlan<sup>a</sup>, Jessica Streets<sup>a</sup>, Philippa L. Jacob<sup>a</sup>, Oska Pugh<sup>a</sup>, Ender Özcan<sup>b</sup>, Simon Woodward<sup>a</sup>

<sup>a</sup> School of Chemistry, University of Nottingham, University Park, Nottingham NG7 2RD, UK

<sup>b</sup> School of Computer Science, University of Nottingham, University Park, Nottingham NG7 2RD, UK

## ARTICLE INFO

### Keywords:

Homogeneous catalysis  
Machine learning  
Cheminformatics

## ABSTRACT

The benefits of using machine learning approaches in the design, optimisation and understanding of homogeneous catalytic processes are being increasingly realised. We focus on the understanding and implementation of key concepts, which serve as conduits to more advanced chemical machine learning literature, much of which is (presently) outside the area of homogeneous catalysis. Potential pitfalls in the 'workflow' procedures needed in the machine learning process are identified and all the examples provided are in a chemical sciences context, including several from 'real world' catalyst systems. Finally, potential areas of expansion and impact for machine learning in homogeneous catalysis in the future are considered.

## 1. Introduction

Machine learning methods are becoming increasingly common in the chemical sciences to efficiently analyse the vast amount of data generated by experimentation and guide scientists to their next reaction [1]. Machine learning has had notable recent success in retrosynthesis [2] and reaction condition prediction [3], advanced by ever increasing computational power and data availability. We herein focus on recent advances in the relatively underexplored field of machine learning for ligand promoted homogeneous catalysis. By focussing on key machine learning concepts, illustrated by examples from homogeneous catalysis, this review serves as a conduit between synthetic chemists and more advanced literature. We build on recent reviews on machine learning in synthetic chemistry [4–11], focussing more exclusively on ligand promoted homogeneous catalysis. We assume basic knowledge of homogeneous catalysis and refer readers to reviews in that area [12–15]. We define here a selective homogeneous catalyst as ML<sub>n</sub>, where M is the activating 'template', most typically a metal (but potentially also an organocatalytic centre, such as H<sup>+</sup>), and L<sub>n</sub> are the reacting and/or controlling ligands through which catalysis is attained. To avoid potential confusion, we abbreviate machine learning (ML) in italics throughout this text.

The opportunities of combining synthetic methods with ML investigation can be illustrated in the application of catalyst screening.

Classical catalyst development approaches typically change only one variable between consecutive reactions (runs) [16]. Such strategies converge only slowly towards improved outcomes. The benefit of ML methods in extracting the features responsible for excellence in catalysis is readily apparent from the following naïve example: a single reaction using a single ML<sub>2</sub> catalyst with a library of five metals and a 20 mono-dentate ligands already provides 1050, i.e.,  $5 \times 21!/(2!19!)$ , pre-catalytic ML<sub>A</sub>L<sub>B</sub> combinations. The number of experimental runs required will be even greater, once other variables (e.g., catalyst activation additives, reaction solvent, temperature, time) are allowed for. An advantage of ML models is they can identify, often rapidly, complicated patterns in multi-dimensional space that the human brain cannot easily envisage. Humans can visualise patterns in two-, three- and even four-dimensional data, but reactions rely on the best combination of a series of variables, and spotting patterns in a matrix of trial runs is challenging. While high throughput experimental (HTE) methods can presently (just about) deal with 'screening' high numbers of experiments (runs), they quickly become expensive in both resources and time as the number of parameters increase. ML methods can augment HTE approaches by reducing the overall number of runs by running fewer initial experiments and predicting the remainder [17].

\* Corresponding author.

E-mail address: [jonathan.hirst@nottingham.ac.uk](mailto:jonathan.hirst@nottingham.ac.uk) (J.D. Hirst).

<https://doi.org/10.1016/j.aichem.2023.100006>

Received 26 May 2023; Received in revised form 29 June 2023; Accepted 10 July 2023

Available online 11 July 2023

2949-7477/© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 2. Machine learning concepts

### 2.1. Introduction to machine learning

Artificial intelligence (AI) refers to the field of computer programs or machines which can mimic human intelligence, such as solving problems and learning. *ML* is a subset of AI, wherein algorithms are used to learn from a dataset and thus make predictions when provided with novel data. A range of *ML* methods is presented in Section 3. Broadly, *ML* approaches can be divided into three categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning methods are supplied with input (training) data, including the target (response) variable of interest. The data are “labelled”, and the output is known for each input (in the context of chemistry, this might be the yield or selectivity to a specific species). Supervised learning infers a general function, mapping the input to output based solely on the training data. This function is evaluated to predict the output when presented with an unseen (new) input. Conversely, unsupervised learning attempts to “automatically” find patterns or associations within unlabelled data (here there is no pre-defined output as a part of the input data). Reinforcement learning is about “acquiring knowledge” for selecting the most appropriate action to take at a given state when interacting with an environment, maximising a notion of cumulative reward over time. Readers should refer to the existing literature on unsupervised and reinforcement learning for examples where these techniques have been applied to chemistry [18–20]. Herein we focus on supervised learning approaches, as they offer the simplest entry to *ML* use.

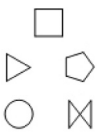
### 2.2. Some vocabulary

Fig. 1 illustrates a generic *ML* process and Table 1 defines some key *ML* terms in supervised learning. After collecting the crude data, these are pre-processed and prepared as input for the *ML* algorithm. The *ML* algorithm produces a function mapping the input to an output through training. The model constructed by the *ML* algorithm is subsequently able to predict the output when supplied with a new unseen instance. Section 3 provides a comprehensive explanation of such *ML* workflows. The interface of chemistry and *ML* has developed its own specialist terminology. It is important to understand this to avoid contextual misunderstandings. Hence, we cover and clarify similar terminologies used in both disciplines in this section.

The simple word ‘optimisation’ is prevalent in both chemical and *ML* literatures, but is generally used in a much more focused manner in the latter. In chemical catalysis ‘optimisation’ indicates a real-world study that has improved a specific reaction outcome: such as yield, selectivity or rate. However, within *ML* literature, ‘optimisation’ is used with its strict mathematical meaning, i.e., finding the best alternative(s) with the maximum or minimum objective value(s), and can be employed within

**Table 1**

Definition of *ML* terms and explanation of symbols shown in Fig. 1.

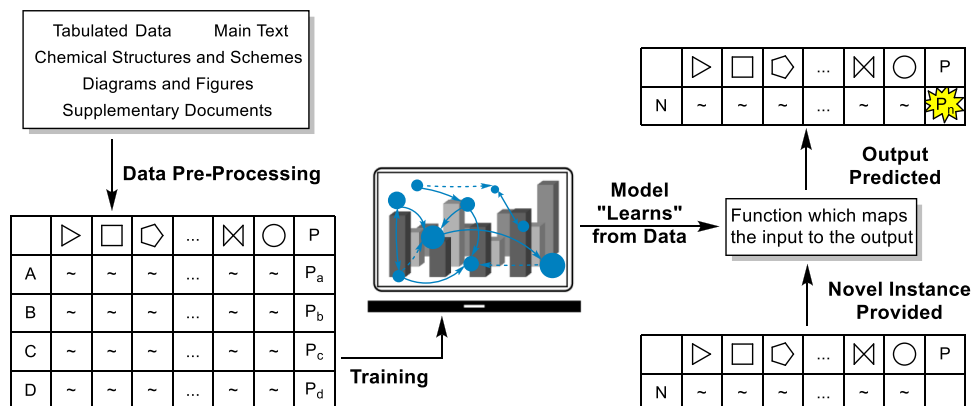
Data Pre-processing	Information regarding catalytic reactions is found in a variety of formats in chemical literature as shown, so must be translated into computer-readable form. Data should also be transformed to a uniform scale (see Section 3.2.2).
	These column headers collectively represent <i>descriptors, features, or input variables</i> (synonyms all used in <i>ML</i> ). These may be real world observables (categorical or numerical), or computationally modelled properties, and typically comprise information about: <i>Reaction Conditions</i> such as temperature or solvent. <i>Reactant Structures</i> represented in computer-readable format. <i>Atomic or Molecular Properties</i> ranging from simple ones like molecular weight, catalyst metal identity, to DFT-derived properties, e.g., ligand HOMO energy. These features are processed by the <i>ML</i> algorithm to predict an outcome (i.e., yields, selectivities, etc.)
A, ~, ~	Rows of values associated with an individual case A, B, etc. (where at least one characteristic of the catalyst system is different) are known as <i>instances</i> . They correspond to the individual catalyst ‘runs’ of classical screening.
X <sub>a</sub> , X <sub>b</sub> ...X <sub>n</sub>	Each <i>instance</i> in supervised learning is associated with a <i>figure of merit</i> , which are also referred to as <i>targets, labels, or output variables</i> . These denote the <i>performance</i> of a catalyst system, such as its yield, selectivity, or rate and are typically attained experimentally.

almost any stage of the *ML* process from data pre-processing to model training and even for model selection [21].

The terms ‘descriptor’ and ‘parameter’ are sometimes presented as synonyms in the catalytic *ML* literature. However, in *ML* the terms are distinct: descriptors represent chemical properties (Fig. 1 and Table 1), whereas parameters are values internally related to the *ML* model that are estimated or learned from the data, e.g., the weights of a neural network (see later, Section 3.4.6). A more general term for a descriptor is a ‘feature’ and this term is also common in both *ML* and chemistry literatures. Finally, another critical term in *ML* is ‘hyper-parameter’ representing an adjustable parameter of an *ML* algorithm, such as the learning rate for training a neural network. The performance of many *ML* algorithms, in terms of predictive accuracy or the efficiency of training, can depend significantly on the settings of the hyper-parameters. Thus, their optimisation (or tuning) is important [22].

### 2.3. An exemplar system for illustrating basic *ML* concepts

*ML* methods use computer algorithms used to detect hidden patterns within data that would not be revealed by human inspection. However, this trait is not so useful when trying to teach *how ML* predictions are achieved when novel instances are supplied (Fig. 1). To make progress



**Fig. 1.** *ML* processes in a supervised learning example. See Table 1 for the definition of *ML* terms and the explanation of symbols.

we will commence our discussion with a ‘toy problem’ for which we already know the answer: alkene ( $C=C$ ) hydrogenation using Crabtree’s catalyst (**2**) [23] and its analogues (Scheme 1). Simple triphenylphosphine,  $PPh_3$  ( $L_A$ ), and pyridine ( $L_B$ ) and suitable iridium sources provide the initial iridium species  $[Ir(COD)(D1)(D2)]^+$  (**1–3**) (where  $M = Ir$  and  $D1$  and  $D2$  are suitable donors, specifically either  $L_A$  or  $L_B$ ). In the following discussion,  $P,N$  refers to a system where one of  $L_A$  or  $L_B$  is phosphine-type ( $PR_3$ ) and the other is amine-type ( $NR_3$ ), and  $P,P$  and  $N,N$  refer to instances where both ligands are phosphine- or amine-type respectively. Literature precedent [23,24] shows that the mixed  $P,N$  system (**2**) leads to the most active (most product formed fastest) catalyst, the  $P,P$  system (**1**) less so, and  $N,N$  ligands lead to very inactive species (**3**). For the rhodium analogues (**4–6**) ( $M = Rh$ ), it is the  $P,P$  system (**4**) that is most active (principally due to decomposition of nitrogen ligated species **5–6** under the hydrogenation conditions). Scheme 1 details five descriptors, selected for their potential influence on activity, which were collected for the organometallic complex: the  $pK_{aH}$  for  $L_A$  and  $L_B$ , together with their molecular weight (mwt), and electronegativity ( $\chi$ ). While a wide range of descriptors could be used for  $L_A$  and  $L_B$ , we use here three simple examples that clearly relate real-world properties of the ligands:  $pK_{aH}$  (correlating to  $\sigma$ -electron density of the donor ligand); molecular weight (mwt, as a very simple measure of ligand bulk); and donor-atom electronegativity ( $\chi$ ) as a moderator of ligand donation potential. See further discussion on descriptors in Section 3.3. The catalytic activity for **1–6**, as estimated from published turnover numbers, is given as the figure of merit (target to be predicted) [23,24]. Given a new ligand,  $L_C$ , the activity of  $[Ir(COD)(L_B)(L_C)]^+$ , **7**, for example, can be predicted based on the regression model, or simply classified as active or inactive (Table within Scheme 1). Our subsequent *ML* discussions will exemplify the process of attaining such predictions.

## 2.4. Classification and regression

Classification and regression are two key types of supervised learning

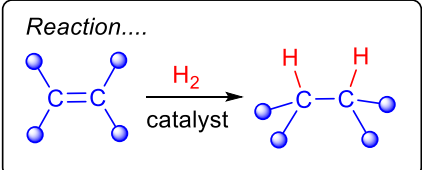
tasks [25,26]. A *classification* model will identify which of two (or more) categories the new input belongs to, whereas a *regression* model will predict a numerical output. In the case of the ‘toy problem’ in Scheme 1, a classification task will categorise incoming data into a series of classes, i.e., “Given  $M(D1)(D2)$ , would the resultant catalyst be active?” Assuming that instead of categories we have numerical values of catalytic activity for each instance in the table in Scheme 1, then, given an unseen instance, the regression task could predict its catalytic rate, i.e., “Given  $M(D1)(D2)$ , what would be its predicted relative rate?” Section 3.4 onwards cover studies using regression approaches. If a larger or more ambiguous dataset had been used from the start, *ML* could be used to reveal that (i) Crabtree’s catalyst  $[Ir(COD)(L_A)(L_B)]^+$  (**2**) hydrogenates hindered alkenes faster than any other member of the library, (ii) the instances sorted into the ‘high’ activity category are commonly the  $Ir/P,N$  catalyst type, or (iii) Crabtree’s catalyst  $[Ir(COD)(L_B)(L_C)]^+$ , a new instance for which catalytic activity may not have been measured, is predicted to have high catalytic activity based on a model trained on **1–6**.

## 2.5. Expectations and limitations of ML

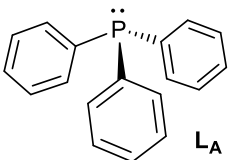
Before discussing *ML* workflows, it is important to appreciate what can be expected from *ML* analysis and its potential limitations. *ML* needs significant data (typically instances equal to at least five times the number of descriptors used) to learn from. It will also predict much more accurately on examples close to the training data than on cases that are distant from what it has learnt. Diverse data are better; discarding ‘poor’ yield/selectivity catalytic run data should be discouraged. Scheme 2 summarises the tools that *ML* novices and their collaborators need to assemble to make a start, and the outcomes that it could be possible to achieve.

The primary requirement is for high quality data [27]. Firstly, the choice of label or experimental data utilised is important. Chemical science’s most commonly collected output - the ‘% yield’ of a reaction -

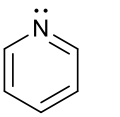
Reaction....



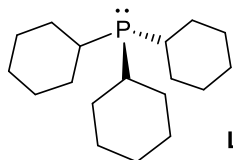
Ligands....



$L_A$

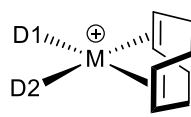


$L_B$



$L_C$

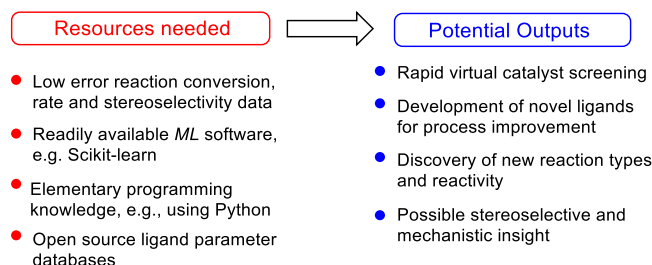
Catalyst library....



	M	D1	D2	pKaH	[M(D1)(D2)(COD)] <sup>+</sup>	donor atom $\chi$	Rel. rate <sup>†</sup>	Active catalyst? <sup>‡</sup>		
				D1	D2	mwt	D1	D2		
<b>1</b>	Ir	$L_A$	$L_A$	5.23	5.23	824.97	2.19	2.19	80	yes
<b>2</b>	Ir	$L_A$	$L_B$	5.23	7.64	641.78	2.19	3.04	5000	yes
<b>3</b>	Ir	$L_B$	$L_B$	7.64	7.64	458.60	3.04	3.04	0	no
<b>4</b>	Rh	$L_A$	$L_A$	5.23	5.23	733.82	2.19	2.19	100	yes
<b>5</b>	Rh	$L_A$	$L_B$	5.23	7.64	550.64	2.19	3.04	30	yes
<b>6</b>	Rh	$L_B$	$L_B$	7.64	7.64	367.45	3.04	3.04	0	no
<b>7</b>	Ir	$L_B$	$L_C$	7.64	9.70	659.92	3.04	2.19	?	?

Indicative relative catalyst activity based on a range of alkenes, using both number (†) and category (‡) approaches

**Scheme 1.** A Crabtree catalyst exemplar: a ‘toy problem’ illustrating the application of *ML* workflows to homogeneous catalysis.



**Scheme 2.** Necessary resources and possible benefits in ML catalysis work.

is often a poor choice. Yield is subject to the vagaries of the isolation, interpretation and reporting processes (e.g., isolated yields versus NMR/Gas Chromatography yields). Measures of reaction selectivity, such as %ee value or some other stereoselectivity indicator, are better choices of output, but these results can also be affected by whether the value was measured before or after purification or isolation [17]. Additionally, stereoselectivity can also be measured directly on crude reaction mixtures. Secondly, bias in datasets is a key problem, for example, not reporting low yield or failed reactions leads to data with few negative examples. Lastly, availability of data, either absence of data, low quality data (without conditions or experimental error), propriety data, or the difficulty in querying databases also can hinder producing a dataset [28]. Recent efforts such as Open Reaction Database [29] have attempted to alleviate these problems, but much work is still required to make data accessible and standardised [30]. The amount of data required for a model varies significantly with the ML method used and the scope of the model; a model built on more (diverse) data will typically be able to cover a larger chemical space as most ML models can only extrapolate to a certain extent. There is no specific rule on the minimum amount of data, with models < 100 instances still providing useful models, if only a modest number of descriptors is used [31]. Thus, the number of experiments in a catalyst screen may be reduced by running several hundred and predicting the remainder.

Although ML can provide accurate and useful results, it can be much more difficult to determine *why* the model has made a certain prediction, especially with so-called “black-box” ML approaches, where outputs can make it difficult to predict what should be the next steps in a catalyst’s development. To overcome this, developing *explainable* and *interpretable* AI is an important and a rapidly developing field [32–34]. To make ML models more interpretable, both the machine learning method and the choice of descriptors should be considered. In general, the simplest model is the easiest to interpret. This may be a model with fewer descriptors or an ML model that is easy to analyse (e.g., a decision tree) [8, 35]. See later sections on descriptors, ML algorithms and case studies for examples of interpretable models. Finally, it is important that the ML research itself is reproducible. Recent guidelines [36] have set out the need for code and data to be accessible and full details of training and validation to be disclosed. To return to Crabtree’s catalyst, to further improve the catalyst for C=C hydrogenation, we would subsequently need to identify or locate the characteristics of the catalytic metal or ligands which are most influential on the activity. This can be achieved by evaluating the relative influence of the descriptors on the predicted activity [37]. This will allow design of improved ligands  $L_A$ , and/or  $L_B$ , that maximise these descriptor(s) ‘information’ (for example by including: M-L bond length or pertinent stereoelectronic properties of the ligand, among many potential examples). This might give an indication of stereoselectivity or mechanistic understanding, but full mechanistic understanding is still largely the domain of quantum chemical studies [38]. By advancing the design process *in silico*, the number of experimental reactions can be reduced, thus saving time and resources.

### 3. Machine learning workflow

#### 3.1. Identifying a chemical problem

ML can be applied to a wide range of problems in chemistry. However, obtaining enough high-quality data is often a challenge in chemistry, as mentioned in Section 2.5. An ideal chemical problem would involve a dataset comprising hundreds to millions of instances, which is potentially easily attainable through literature mining, HTE or computational methods. The instances should include a wide range of values to encompass the breadth of the problem. A model must know about failures, successes and the range in between to deal effectively with unseen instances (i.e., new catalyst formulations). It is important that the data be drawn from the underlying distribution so that patterns, perhaps arising from related structures or the presence of certain functional groups, can be learned. Simultaneously attaining all the above criteria is often difficult. Data may be limited, due to instances being expensive to generate, and literature datasets can frequently be skewed to a (false) positive outcome or can be encoded in unhelpful/non extractable way. However, there are ways of overcoming these challenges [39].

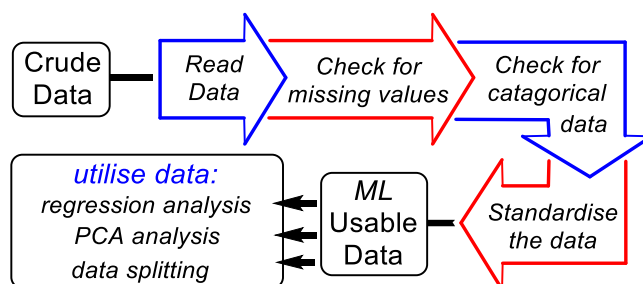
#### 3.2. Data preparation and pre-processing

Data preparation and pre-processing takes raw “real-world data” and converts it into a machine-readable format, suitable for presenting as input to an ML computer program. There are many techniques for data preparation and pre-processing, including data cleaning, scaling, data balancing, dimensionality reduction, instance reduction, and more. See subsequent sections for examples of data preparation and pre-processing. Often, it is not necessary to apply all those techniques while tackling a problem, which would ultimately depend on the data being used. In this section, we concisely explain some of the selected techniques. Fig. 2 represents a typical approach to data pre-processing and use.

##### 3.2.1. Issues with data

The first step is to collect the data from experiments or the literature. It is imperative to ensure that the data is error free. For example, in a stereoselective reaction, it should be checked that the enantiomer of the product, and any ligand used, has been correctly assigned. Data should be collected into a database or other suitable machine-readable form such as comma separated values (.csv) file. Care should be taken to ensure units are consistent. In a real-world raw dataset, there are frequently issues to be resolved, such as missing descriptors and improper data types. Therefore, checking the data initially is important [40], for example, in a ligand screen, there may be missing runs. A trivial fix for missing descriptors would be dropping the relevant instances and/or descriptors containing null/empty values, which can cause loss of valuable information. There are also *data imputation* methods based on machine learning and optimisation, which can generate substitutes for the missing data values [41,42].

It is important that the data types are appropriate and consistent for each descriptor within the dataset. For example, if there are categorical



**Fig. 2.** Indicative data pre-processing and use.



descriptors, e.g., a category of ligand size, that can be ordinal (that can be ordered), such as “small” < “medium” < “large” (e.g., they can be processed and encoded as 1 < 2 < 3). Alternatively, the descriptors may be nominal, i.e., cannot be ordered, such as catalyst or reagent form (“solution”, “solid” or “unknown”). A pre-processing phase would involve ensuring that the categorical data for the descriptor are labelled appropriately.

### 3.2.2. Feature scaling

Feature (descriptor) scaling or transformation is a process which ensures different numerical variables within the dataset are all on the same scale, which allows each variable to be considered by the ML method equally [43]. For example, %selectivity falls on a scale of 0–100, whereas a temperature range within a study could be far wider, or narrower. Without adjustment, these two features would not be treated comparably, potentially leading to a bias within the analysis [44]. Where an ML algorithm is heavily affected by the range of a descriptor (i.e., the larger the range of a descriptor the larger its influence on the calculation), scaling is essential. Other ML algorithms that construct models without using a distance metric may eliminate the importance of descriptor scaling. (See later: distance-based algorithms including k-nearest neighbours (k-NN), support vector machines (SVM) and neural networks; models that do not use distance metrics include decision trees and random forests).

The main descriptor scaling methods are *standardisation* and *normalisation* [45]. The standardisation process computes the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the values ( $X$ ), and then from each data entry the mean is subtracted, and the resultant value is divided by standard deviation producing a substitute value,  $X'$  (Eq. 1). The normalisation method, also known as max-min scaling, finds the minimum ( $X_{\min}$ ) and maximum ( $X_{\max}$ ) values for the descriptor, then subtracts the minimum from the data entry dividing the resultant value by the difference between the maximum and minimum values for scaling (Eq. 2).

$$X' = \frac{X - \mu}{\sigma} \quad (1)$$

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (2)$$

Both processes are highly sensitive to outliers, as skewed input data will lead to skewed scaled data. Several other scaling techniques, out of the scope of this overview, can deal with outliers. Descriptor and instance selection (i.e., excluding a ligand or descriptor extremely different to the bulk) can also be useful.

### 3.2.3. Data balancing

Handling imbalanced data (in which some crucial aspects are underrepresented and/or skewed) which influences the ML performance is a challenging issue in supervised learning. For example, the number of instances in one class (e.g., low temperature catalyst runs) might be significantly fewer than other classifications (e.g., medium to high temperature runs). This could be a problem because ML algorithms often attempt to maximise accuracy and minimise error in predictions, and they may simply predict that all instances belong to the majority class. The performance of an ML algorithm can be improved through data balancing, for example via sampling. *Oversampling* augments the original dataset by generating artificial data to add under-represented data. One of the commonly used techniques is SMOTE (Synthetic Minority Oversampling Technique) for classification and its variant SMOTER that adopts SMOTE for regression [44,46]. *Undersampling* is the opposite strategy, where instances are removed from the dataset. Singular metrics to evaluate the performance of ML algorithms (i.e., reduced to a single value such as accuracy and error) can be sensitive to imbalanced data. Performance metrics are discussed in Section 3.4. Hence, the choice of the performance metric is important while applying a data balancing technique. He et al. and Branco et al. cover various techniques for data

balancing along with potential issues and relevant informative performance metrics [44,46].

### 3.2.4. Dimensionality reduction

An important characteristic of a dataset is its dimensionality, i.e., the number of descriptors it has. Dimensionality reduction is concerned with mapping the data from this high-dimensional space to lower-dimensional space, whilst maintaining the significant patterns in the initial data. Reducing the number of dimensions in the data enables simplified models to be used and patterns in the data to be more easily spotted. Additionally, overfitting [47], where models are trained too tightly to the detriment of new test data, can be avoided. This can also allow faster training. Without such processes, as more descriptors are added, the number of data instances required by the model grows too rapidly for accurate predictions to be made – the so-called ‘curse of dimensionality’ [48]. Feature selection and feature projection can be used to reduce the dimensionality of a descriptor set (Scheme 3).

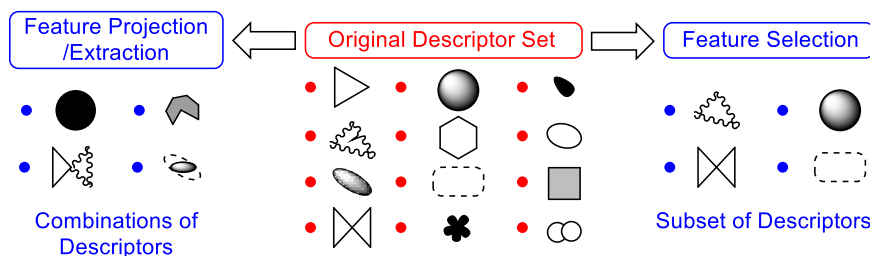
*Feature selection* is a form of dimensionality reduction for choosing a subset of relevant descriptors for ML methods [49]. For applications in ligand promoted homogeneous catalysis, the influence of some features might not be sufficiently obvious for the selection(s) to be done manually. A more programmatic approach would be to use *variance thresholds*, whereby the variance of each descriptor is computed and if that value does not exceed the specified threshold, then the descriptor is removed, on the basis that a descriptor with limited variation will have limited predictive power [50].

Feature selection methods can be categorised as *filter*, *wrapper*, or *embedded*. *Filter* methods [51] are used in the pre-processing stage and are independent of any model built. A subset of descriptors is selected, based on the relationship of each descriptor with the target variable, based on a metric such as correlation. *Wrapper* methods [52] perform a search of descriptors guided by the performance of the subsequent model built. Either an initial model is built with a small subset of descriptors, and more descriptors added (forward selection), or descriptors are removed from an initial model containing all descriptors (backwards elimination). *Embedded* methods [53] are more advanced and select descriptors while constructing the model based on fitting errors, and their hybrids.

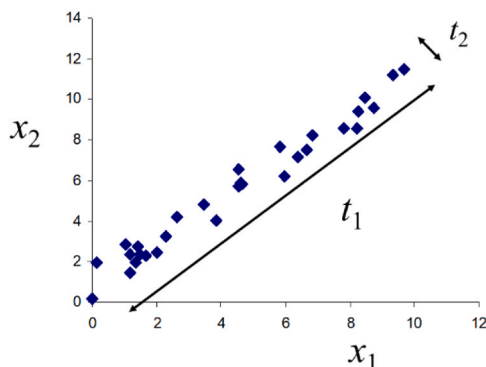
*Feature projection* (also called *feature extraction*) is another approach to dimensionality reduction, where the original descriptors are used to generate a smaller set of descriptors that define the instances within the data set more usefully (and succinctly) than the original ones. This technique moves the data with high dimensionality to a space with fewer dimensions. There are various feature projection methods which include Linear Discriminant Analysis (LDA), Independent Component Analysis [54], Autoencoder, Random Projection [53], Principal Component Analysis (PCA), and Partial Least Squares (PLS), and more. In this section, we explain the latter two well-known methods.

PCA [55,56] takes the original data and creates linear combinations (called principal components) of the descriptors and projects them on to a set of orthogonal axes (Fig. 3). The principal components are ranked in order of importance based their variance to summarise the original data distribution optimally. Optimally in this case means maximising the variance of the principal component.

The first principal component shows the greatest variance in the data. PCA preserves large pairwise separations better than smaller ones. In other words, instances that are well separated in the original descriptor space will still be well separated after the PCA. PCA is an unsupervised method. It does not distinguish any labelling of the data and only identifies the variance of the descriptors, which could lead to data loss. The cumulative variance retained gives an indication of how much data has been preserved. If the dataset is reduced too much, underlying trends in the data towards the target value may be lost. Although it is trivial to see the influence of each descriptor on each principal component, it does make subsequent analysis of the model harder as the original descriptors are not directly used, which makes it



**Scheme 3.** To reduce the dimensionality of a descriptor set, typically feature selection and feature projection are employed. In feature selection, a subset of the original descriptors is chosen (represented by the shapes selected from the original descriptor set). In feature projection/extraction, the original descriptors are combined to form a new set of descriptors (represented by the hybrid shapes).



**Fig. 3.** Principal components,  $t_1$  and  $t_2$ , are linear combinations of the original descriptors,  $x_1$  and  $x_2$ , and correspond to the definition of a new set of axes in which  $t_1$  accounts for the greatest variance.

harder to *explain* the predictions based on the descriptor importance or other analysis.

The Partial Least Squares (PLS) method [57] is applied to both the descriptors and the target variable. Latent variables,  $t$ , are linear combinations of descriptors, and the target variable,  $y$ , is predicted based on a linear combination of the latent variables:

$$y = a_1 t_1 + a_2 t_2 + a_3 t_3 + \dots + a_n t_n \quad (3)$$

where the coefficients  $a$  are weighting coefficients. Usually only the first few latent variables are used, with the number determined through a cross-validation approach. PLS serves to reduce the dimensionality of the dataset by aggregation of collinear descriptors. This is similar to PCA (see Fig. 3 and above), but the target variable is considered. In particular, PLS defines latent variables which maximise the covariance of the descriptors and the target variable. PLS is most effective on datasets with high collinearity, as this enables the reduction of the data into a handful of descriptors that account for the majority of the variability in the target variable. A poor ratio of data to descriptors is a common issue in chemistry that has made PLS a popular method in the field [58,59]. It has the same drawbacks as PCA in data loss and linear modelling.

There are many other dimension reduction techniques beyond PCA and PLS. One example is Uniform Manifold Approximation and Projection (UMAP), which has found particular use for quickly mapping chemical space. While the details of this algorithm are out of the scope of this review, we refer readers to the following references [60,61].

### 3.2.5. Data splitting and validation

The primary goal of a supervised ML technique is to train a model in order to predict a desired characteristic of previously unseen test examples correctly. *Holdout* is the simplest approach to data splitting. Partitions for training and testing is achieved through random sampling of instances. The size of the subset of data for training is usually larger than for testing (e.g., 70:30 or 80:20). Achieving high training accuracy

alone is not fully useful, if the prediction is not accurate enough based on the test data. Large differences in predictive accuracy on training and test data of unseen instances indicate that the model is *overfitted* (often caused by too many descriptors for the number of instances). A well-established protocol to estimate the presence of such errors is to apply cross-validation, in which the dataset is partitioned into multiple subsets, a part of which is used for training and the rest for validation and testing.

The most common cross-validation approaches are  $k$ -fold cross-validation and leave-one-out cross-validation [62]. In the former, the dataset is split into  $k$  subsets and, sequentially,  $k - 1$  of them are used for training and the remaining one for testing. This approach is repeated  $k$  times, until all subsets are used once for testing. *Leave-one-out cross-validation* involves using a single instance in the dataset as test data and the remaining instances as training data, until all instances are used for testing once. In other words, the procedure is the same as in  $k$ -fold cross-validation, but in this case,  $k$  is the number of instances in the dataset. However, this requires large computational power and time when the dataset is large. If the hyper-parameters of an ML algorithm require optimisation, a nested cross-validation approach is usually applied, whereby the hyper-parameters are tuned based on a validation subset of data. Then the tuned final model is evaluated on the test set to ensure the integrity of the estimate of the predictive accuracy of the model [63].

### 3.3. Descriptors

Descriptors represent characteristics of a chemical entity in a form suitable for ML algorithms, so that they can be used to create models. Molecular descriptors themselves may be characterised by the dimension of their representation of the molecule (1D–4D), or by their nature (constitutional, topological, geometric, electrostatic, and quantum mechanical), although these groupings overlap. One might anticipate that the more sophisticated the descriptor, the more predictive the model. However, this is not always the case and working upwards from the simplest descriptors often provides insights.

1D descriptors contain global information about the molecular structure, for example, atom or fragment counts, molecular weight, etc. Their simplicity means they can suffer from degeneracy and the identity of isomeric compounds are lost. Models based on 1D descriptors can provide a useful baseline approach and such descriptors can contribute to predictive models in combination with other descriptors. 2D (bond connectivity) descriptors are very common [64], with free and commercially available programs able to generate many different types with minimum computational time. They are derived from chemical structures and describe connectivity and structure within molecules. The number of bond types or counts of aromatic rings can be included, analogous to topological descriptors. The distance between two coordinating atoms in a ligand would be an example of a 2D descriptor. 3D descriptors rely on the generation of 3D structure via molecular mechanics or ab initio calculations [65]. Depending on the level of theory, they may be time-consuming to calculate. Examples include the

principal moment of inertia, representations of the solvent accessible areas or van der Waals volume [66]. Molecular interaction fields, popularised in the Comparative Molecular Field Analysis (CoMFA) technique [67], also find utility in 3D ML studies of catalysis [68]. Finally, 4D descriptors take conformational changes over time into account by taking the average of conformers generated over the course of a molecular simulation. 4D descriptors are very computationally expensive to calculate, but the information derived is complex and detailed [69,70].

Experimental descriptors are also common. However, they can be expensive if they can only be obtained by time-consuming experimentation or lengthy literature searches. However, these descriptors themselves can be often predicted. The Hammett  $\sigma$  constants measure the electron-donating and -withdrawing power of organic substituents. They were traditionally calculated from experimental ionisation energies [71]. However, there has been recent effort to directly calculate these parameters for new substituents without the need for additional experimentation [72]. Another example are descriptors developed by Tolman [73], such as his experimentally measured Tolman Cone Angle (TCA) which describes the steric bulk of ligands. More recently, TCA was calculated from quantum-mechanically optimised 3D ligand structures and used to predict catalytic activity [74]. Other descriptors which may be experimentally derived or calculated are  $pK_a$  [75] and electronegativity [76].

The interpretability of models can also be improved through the choice of descriptor [77]. A small number of “hand-crafted” descriptors, which chemists intuitively understand, can result models which are easily to analyse and justify. One study advocated for spectroscopic descriptors which were physically meaningful, cheap and extensive [78].

### 3.3.1. SMILES and InChI

*Simplified Molecular Input Line Entry System* (SMILES) is a line entry format for chemical structure notation [79]. SMILES is a workhorse in cheminformatics, providing a compact string representation of molecules which is machine readable. There are various utilities for converting SMILES into two- or three-dimensional representations of the molecule [80]. The simple rules for constructing a SMILES string are given in Table 2.

For example, the SMILES representation of the molecule benzoic acid is ‘c1ccccc1C(=O)O’, combining the rules from Table 2 for the aromatic benzene ring and carboxylic acid; the number of hydrogen atoms required to satisfy valency are usually implicit. Stereochemistry can be shown in SMILES, and each SMILES uniquely represents a molecule. However, depending on which atom the SMILES begins with, a molecule may have many valid SMILES. For full SMILES rules refer to the established guidelines [79].

*International Chemical Identifier* (InChI) is another string representation of molecules [81]. Benzoic acid is denoted 1 S/C7H6O2/c8–7(9) 6–4–2–1–3–5–6/h1–5 H,(H,8,9). ‘1 S’ defines the version number, ‘1’, and ‘S’ refers to the fully standardised version of the InChI (StdInChI). ‘C7H6O2’ is the empirical formula, the next section defines which atoms are connected, and the final section denotes which atoms have hydrogen atoms. Most of an InChI string is human-readable and they can include information about tautomeric state and stereochemistry. Unlike SMILES,

**Table 2**

Basic rules for constructing a SMILES string.

Molecular feature	Representation
Non-aromatic atoms	Element symbols, first letter in upper case letters
Single bonds	- (but may be omitted)
Double bonds	=
Branching	Denoted using ()
Ring closure	Label (with the same number) atoms that were connected to each other
Aromatic atoms	Lower case letters

**Table 3**

Important definitions to assess a classification model. \*Real refers to the experimental or calculated target value in chemistry (e.g., catalyst is active/inactive) which is assumed to be correct.

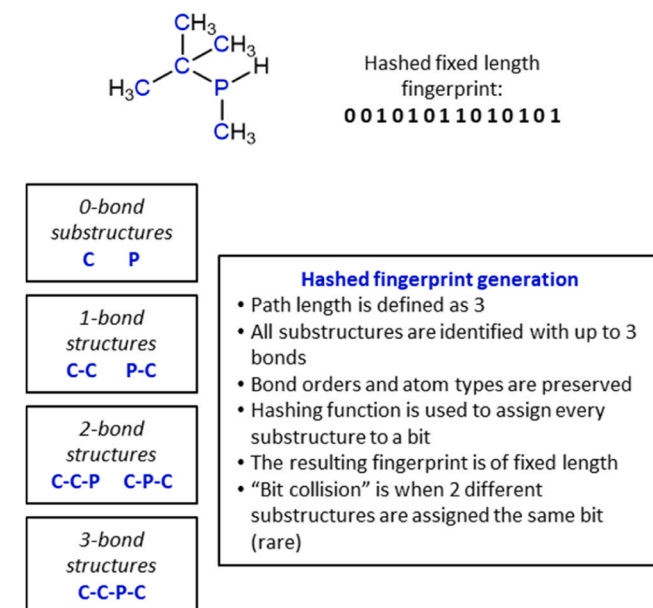
Term	Prediction Value	Real* Value
True positive (TP)	Positive	Positive
False positive (FP)	Positive	Negative
True negative (TN)	Negative	Negative
False negative (FN)	Negative	Positive

every chemical structure has a unique StdInChI string, which is useful if isomeric compounds appear in the data set. InChI forms are long and of non-fixed length for large molecules, which can handicap comparing the similarity of two structures. StdInChIKey, an algorithm generated (non-human readable) fixed-length string facilitates storage, but cannot be directly converted back to the chemical structure [82].

### 3.3.2. Fingerprints

Molecules are often represented by chemical ‘fingerprints’ to aid substructure and similarity searches [83,84]. These fingerprints are of a pre-defined length (though the actual length depends on the fingerprint used) and broadly fall into two categories: structural key based and hashed based fingerprints. Structural key fingerprints (such as MACCS [85] and PubChem [86]) encode predefined structural features, such as a substructure or fragment, as a ‘1’ or ‘0’ depending on its presence or absence. Alternatively, a more flexible approach is to enumerate through fragments up to a certain size. This enumeration can be path-based as in Fig. 4, where a connectivity of up to three has been included (e.g. Daylight fingerprint) [87], or circular by considering the environment of an atom by its ‘radius’ or ‘diameter’ (e.g. extended-connectivity fingerprints) [88]. To standardise between different sized molecules, these fragments are converted into a bit vector using a hashing function. The hashing function converts the fingerprint from an arbitrary length to a fixed length. There are numerous other types of fingerprint, such as 3D pharmacophore fingerprints [89], not covered in this overview.

Fingerprints are easy to use for molecular similarity and substructure searching using a variety of similarity coefficients and distances. As the fingerprints do not encode the full structure of a molecule, they cannot be easily converted back to the full structure. Fig. 4 visualises the overall



**Fig. 4.** The process of generating a fingerprint based on path length [83,88].

procedure using a phosphine,  $\text{PHMeBu}^t$ , which has been used (in its  $\text{BH}_3$  protected form) for chiral ligand synthesis [90].

### 3.4. Machine learning methods

ML models have a trade-off between *bias* and *variance* [1]. Bias is the systematic error within the ML model itself and can be defined as how well the model matches the training data. A highly biased model may be too general, not fitting the data particularly well, and low bias refers to a near perfect fit. The variance assesses the difference in fit between different portions within the training data: if the fits for all portions are similar then the model is said to have low variance. However, if they are wildly different the model is considered to be overfitted, especially if the performance on unseen test data is poor. This normally occurs when there are too few instances for the number of descriptors used (although there is no consensus on the number of instances per descriptor) [91,92]. A good model is one with low bias (sensitive to training data), and low variance (performs well with new data). However, there is often a trade-off between bias and variance to get the most suitable model.

There is often no clear way of determining which ML method to apply to a dataset up front. Almost every ML method has been applied to homogeneous ligand catalysis (see subsequent examples) and the choice of method depends on the structure and type of input data and the amount of training data. Usually, neural networks are preferred for large datasets with SMILES or graph inputs [93,94]. In practice, many machine learning methods will be trialled for a given dataset, and the best method retained.

#### 3.4.1. Model performance analysis

Various metrics to evaluate the performance of ML algorithms allowing analyses of their predictive capability, tuning and optimisation of their parameter/hyper-parameter settings, and comparison of different ML models. It is useful here to use the toy problem of Scheme 1 where a classification problem requires identification of which ligands give active or inactive catalysts. For the following discussion several definitions are required. In this example, *positive*, denotes that the catalyst is active and *negative* that the catalyst is inactive.

A commonly used metric for evaluating a classification model is *accuracy*, indicating the fraction of correct predictions out of the total made. It is an indication of the overall accuracy of the model but may be affected by the number of each class in the dataset. Suppose a dataset with 90 active catalysts and 10 inactive catalysts. A (very poor) model which assigns all as active would give an accuracy of 90%, which is misleading.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total predictions made}} \quad (4)$$

*Precision* is the ratio of true positives and total positive predictions. This indicates how good the model is at positive predictions (active catalyst predictions). In the above example the model would also have 90% precision.

$$\text{Precision} = \frac{TPs}{TPs + FPs} \quad (5)$$

*Recall* or *sensitivity* is the ratio of true positives and total positives in the dataset [95]. This indicates how good the model is at identifying positive examples (active catalysts). In the above example the model would have 100% recall since all active catalysts were assigned as positive.

$$\text{Recall} = \frac{TPs}{TPs + FNs} \quad (6)$$

*Specificity* is the ratio of true negative predictions and total negatives in the dataset [95]. In the above example, the model would have a specificity of 0% since no inactive catalyst was correctly identified. Thus, it is crucial to calculate multiple metrics to get the true assessment

of a model.

$$\text{Specificity} = \frac{TNs}{TNs + FPs} \quad (7)$$

Fall-out rate and miss rate are other statistics describing the false positive and false negative rates respectively. These, and more advanced statistics such as F1 score and the “Area Under the Curve” (AUC), are not covered here.

For a regression problem, the root mean square error (RMSE) of the prediction is a well-established measure of success. It is defined as the square-root of the sum of the difference between the real and (y) predicted ( $\hat{y}$ ) values, squared and divided by  $n$ , the total number of instances. The mean absolute error (MAE) provides information on the likely error in each prediction. It is defined as the sum of the difference between the absolute values divided by the number of instances.

$$\text{RMSE} = \sqrt{\frac{\sum_i^n (y_i - \hat{y}_i)^2}{n}} \quad (8)$$

$$\text{MAE} = \frac{\sum_i^n |y_i - \hat{y}_i|}{n} \quad (9)$$

Outliers adversely affect these metrics, since they are based on the mean. Therefore, careful analysis of the errors is required. The standard deviation of the target variable can be compared to the average error to give an assessment of the predictive power of the model. If the error is less than the standard deviation the model is performing better than a naïve assignment of the mean target variable to the test set [96].

ML methods should be compared based on an appropriate metric using an appropriate statistical test, which can establish whether one prediction is significantly better than another. The statistical tests usually define a null hypothesis that the results of two predictions for different ML models are not significantly different and, based on the outcome of the test, the hypothesis is either accepted or rejected. For details of *parametric* and *non-parametric* statistical tests, we direct readers to the following references [97–99].

Some selected common ML methods used for regression and classification are covered in the following sections.

#### 3.4.2. Linear regression

This simple method assesses the correlation between a descriptor and the target variable using a line of best fit. The line is fitted using the least squares method, where the sum of the squares of the residuals (SSR) is minimised. The equation of the line is given as:

$$y = mx + c \quad (10)$$

where  $y$  is the target variable,  $x$  is the descriptor, and  $m$  and  $c$  (the gradient and intercept) are optimised by the model. This line minimises the vertical distance between each of the points from the real data and the fitted line. For multidimensional problems, multilinear regression (MLR) [100] using the same technique is applied, but with as many coefficients as descriptors to describe the associated hyperplane. For a dataset with two descriptors,  $x_1$  and  $x_2$ , and the target variable,  $y$ , the equation of the hyperplane would be:

$$y = ax_1 + bx_2 + c \quad (11)$$

The coefficients  $a$  and  $b$  describe how positively or negatively that the associated descriptors contribute to the prediction and to what extent.

The fitted coefficients allow easy analysis of the contributions of each descriptor, giving highly interpretable models. MLR cannot deal with many collinear (correlated) descriptors. However, dimension reduction can be used to reduce the dimensionality of the dataset (see previous section on PCA and PLS). Additionally, non-linear relationships cannot



be modelled, so MLR may not be suitable for more complicated relationships. Due to its transparency, MLR is still popular for machine learning for homogeneous catalysis, especially for smaller datasets [101]. However, one of the more advanced methods described in the subsequent sections typically outperforms MLR for larger datasets, as seen in the majority of case studies presented in Section 4. For a case study using this model see Section 4.1.

### 3.4.3. Decision trees

This method [102] encodes a series of binary choices to make decisions and classify the inputs into ever similar groupings. They are able to handle a wide range of data types such as binary, numeric, ranked and even multiple-choice data in a single tree. Trees begin with a root node. The root is defined as the descriptor that contributes greatest variability to the target variable and is determined by calculating an impurity score: a measurement of the probability that a classification will be incorrect. This same process is followed to define the internal nodes and finally the leaf nodes where the prediction is made. A decision tree is shown in Fig. 5 for the Crabtree catalyst exemplar problem described in Scheme 1. Data are partitioned according to the presence of Ir and descriptor values. Given new complex 7 (Scheme 1), the relevant descriptor values could be used to make a prediction. Fig. 5 would therefore predict a high relative rate of 5000 for complex 7. An important consideration is the maximum depth of the tree. In Fig. 5, due to the small descriptor space, the maximum possible depth has been reached where each instance has been separated into its own leaf node. A single decision tree is rarely used in machine learning since the error is typically too high. Instead, an ensemble of decision trees, known as a forest, can be used.

### 3.4.4. Random forest

This powerful ensemble method [103,104] combines many individual decision trees to make a prediction based on the aggregated results. The random forest algorithm provides a number of benefits, including: (i) mitigation of overfitting through the use of multiple decision trees for prediction, (ii) higher accuracy, (iii) ability to handle large and/or high dimensional data, and (iv) estimate missing data maintaining accuracy [105].

The ‘random’ factor in this approach is a consequence of how the decision forest is grown. Firstly, only a randomly chosen subset of instances are used to grow each tree. Secondly, rather than using all descriptors at each decision point (node), only a random subset of descriptors is used. These steps are repeated hundreds of times until a forest of decision trees (of differing predictive ability) has been grown.

Predictions are made using every tree in the forest and these predictions are aggregated. A new instance to be predicted is applied to every tree and the prediction is a combination of the entire forest. The aggregation of many results often gives the best prediction with low

variance (c.f. the ‘wisdom of crowds’). Depending on the type of data, the aggregation can be done using a voting system for categorical data or by taking the average for continuous data.

Random forests have several hyper-parameters including the minimum sample size and maximum depth. The minimum sample size is the minimum number of samples required to make a split. Typically, the tree is grown to until all terminal nodes contain less than the minimum sample size. This is the tree depth. Additionally, the number of trees in the forest can be varied. Although default hyperparameters generally give good predictive models, they can be optimised to make them even better [106,107]. An additional benefit is that the relative importance of each descriptor can be estimated. This makes the models more interpretable since highly weighted descriptors could give insight into the catalysis system under study. Due to their ease of use and relative insensitivity to hyper-parameters, random forest is a popular method for machine learning in homogeneous catalysis [37,108]. For a case study using this model see Section 4.2.

### 3.4.5. *k*-nearest neighbours algorithm (*k*-NN)

In this method [109] all inputs, including the descriptor values and labels, of the instances for training are stored. For classification, when an unseen instance is fed into *k*-NN as a new input, the algorithm returns the class label which is the most frequent among the *k* nearest training instances to that input based on a distance metric as illustrated in Fig. 6. A commonly used metric is the *Euclidean distance* for measuring the pair-wise proximity of the instances in the descriptor space. Similarly, for regression, *k*-NN returns the average value of *k* nearest neighbours.

Since *k*-NN makes predictions based on a distance metric, normalising/standardising the data is crucial. The simplistic nature of the algorithm may make *k*-NN unsuitable for datasets with a large number of descriptors. Each descriptor is necessarily treated with equal weight in the distance measurement. Therefore, unbalanced descriptor sets may give predictions based on the dominant features represented (e.g., too many ligand descriptors rather than metal descriptors). In addition, *k*-NN is poor at extrapolation, since predictions are based directly on distance to known training examples. For these reasons, there are limited cases of *k*-NN used for homogeneous catalysis datasets but they have been used in combination with other methods [109]. Much like MLR, *k*-NN excels in its simplicity and transparency. It can also be used as a benchmark to compare more complicated ML methods to assess the utility of these models [110]. Lastly, there are several modifications to *k*-NN, which can increase its accuracy. We refer readers to the following reference [111].

### 3.4.6. Support vector machines (SVMs)

An SVM works by dissecting data using a hyperplane which lies in the centre of two margins defined by the support vector classifiers [112, 113]. Kernel functions are used to transform non-linear data so a linear separation may be found, as shown in Fig. 7.

Kernels, functions that compare instances to each other, are used to find patterns at a higher dimension than the data naturally allows. A kernel function,  $K(a,b)$ , works by calculating the dot product of two

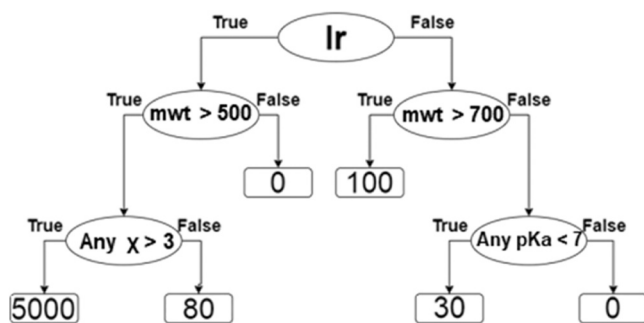


Fig. 5. Decision tree for the Crabtree catalyst exemplar described in Scheme 1. Branch points correspond to the presence or absence of iridium and thresholds on the descriptor values; values at the leaves are relative rate. Due to the size of the exemplar data the tree is grown to its maximum depth. Otherwise, the mean values of every instance in the leaves would be taken as the prediction for this regression problem.

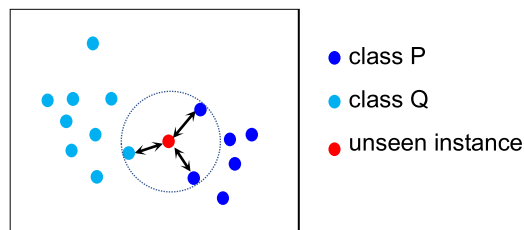
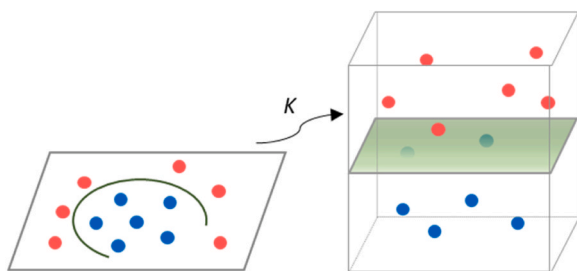


Fig. 6. 2D scatter plot of the input (training) instances based on two descriptors, belonging either class P or Q. Applying 3-nearest-neighbour algorithm would classify the unseen instance as P, since two (majority) of the three nearest neighbours belong to that class.



**Fig. 7.** The kernel function,  $K$ , transforms the data from the input space (left) to a higher dimensional feature space (right) where linear boundaries can be found.

instances,  $a$  and  $b$ , essentially the magnitude and direction between the vectors derived from the instances. Taking the polynomial kernel as an example where  $d$  is the degree of the polynomial and  $r$  as the coefficient, if  $r = \frac{1}{2}$  and  $d = 2$ :

$$\begin{aligned} K(a, b) &= \left(ab + \frac{1}{2}\right)^2 = \left(ab + \frac{1}{2}\right) \left(ab + \frac{1}{2}\right) = ab + a^2b^2 + \frac{1}{4} \\ &= \left(a, a^2, \frac{1}{2}\right) \cdot \left(b, b^2, \frac{1}{2}\right) \end{aligned} \quad (12)$$

The dot product indicates the relationship between two instances by providing the coordinates of them in 2D space so defining the position of each instance relative to all others. The kernel trick (Fig. 7) is this principle: analysing the data as if it were in higher dimensional space rather than performing the actual transformation in order to define a hyperplane.

The margin can vary in ‘hardness’ depending on the separability of the data: a soft margin accommodates some misclassifications, but a hard margin has no tolerance. To increase the bias and reduce the variance, a soft margin is preferable to the instances that lie in this region; these instances are categorised as the support vectors. This can be applied to a 1D linear dataset where the threshold and support vector classifiers are simply points on a line, and can be extended to  $n$ -dimensional data, where hyperplanes are defined by  $n-1$  dimensions. For non-linear data where no single threshold can be defined, an SVM can use a kernel to effectively transform the instances onto higher dimensions until a single hyperplane can be fitted. Kernels are functions that output a similarity matrix containing values for every instance compared to every other instance in the dataset.

Computational costs are kept low by avoiding the transformation and instead providing the similarity matrix. As with all these methods, it must be refined to suit the dataset. Various kernels are available, such as the radial basis function, which behaves like a weighted nearest neighbours model, clustering based on the proximity of other instances. The computational power of individual kernels is nominally dependent on the dataset. However, to systematically find the best kernel and optimise the hyper-parameters for each takes time, as models are highly dependent on the hyper-parameters used. In fact, it is challenging to optimise the hyper-parameters even when the kernel is fixed to be the radial basis function [114]. SVMs are suitable for the small and medium size datasets typically seen in chemistry [115,116]. However, a disadvantage to SVM is that the computational time to select the support vectors and construct the hyperplane increases exponentially as the training data size increases [117]. For a case study example of use of this model see Section 4.2.

### 3.4.7. Neural networks: from perceptron to deep learning

Inspired by early attempts mimic the activity of neurons of human brains, these use multiple decision layers connected by binary switching functions [118]. Neural networks are computationally expensive and time consuming to train and result in predictive logic that is often opaque to human comprehension and tricky to rationalise. However,

with enough time and data they can provide highly accurate results.

The artificial neuron (node) is a simple computational unit which processes input values, taking the weighted sum of input values, transforming/scaling that sum through a non-linear function mapping to an output value. The perceptron is the simplest neural network architecture that employs supervised learning for binary classification. Fig. 8 illustrates a single layer perceptron that is composed of multiple input neurons passing on the input signal/values directly and a single neuron in the output layer. Each connection between neurons carries a weight (corresponding to synapses in the biological analogy, e.g.,  $w_1, w_2, \dots, w_n$ ).

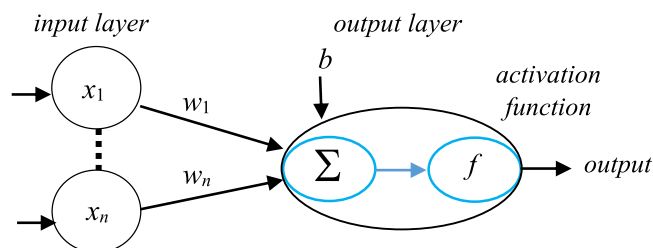
The weighted sum of the input values received from the preceding input layer, including any *bias* function,  $b$ , is fed into the activation function. The activation function decides whether a neuron *fires* or not. Usually a step function (Eq. 13) is used as an activation function which ensures that the output is either 1 (indicating firing of a neuron) or 0. The *bias*,  $b$ , in the activation function adjusts the *threshold* away from origin.

$$f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^n w_i x_i + b > 0 \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

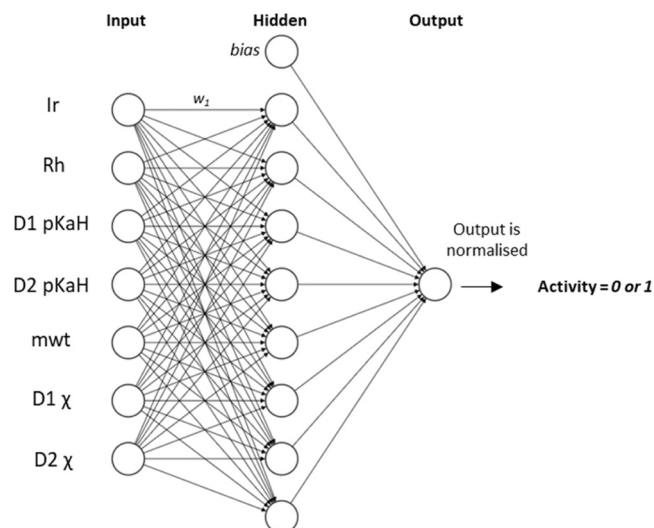
During the training, optimal weight coefficients are computed automatically using the learning algorithm which iteratively compares the output to the predicted output propagating the error back and updating the weights until that error reduces to a satisfactory level.

One of the simpler types of neural networks is the multi-layer perceptron (see Fig. 9). There are several layers of nodes: input, output, and hidden; there may be many hidden layers. In the fully connected version, each neuron (node) in one layer is connected to every other neuron in the next layer. The weights and biases of the connections in the neural network are updated by the *back-propagation* of errors algorithm [119], where a network is iteratively updated by running training data forward through the model, then updating the weights to minimise the overall error.

The hyper-parameters of a neural network include the number of hidden layers and the number of neurons (nodes) in each hidden layer. There is not an obvious way to decide the optimal number of hidden layers and nodes, which can lead to under- or over-fitting. However, there are several guides on network architecture [120]. Large neural networks may also require long training times or specialist hardware (e.g., GPUs) for large datasets. Although the advent of tools such as TensorFlow and PyTorch have made neural networks more accessible, there is still a significant barrier, not least due to the sheer number of options available, for new users. Neural networks have gained popularity due to their ability to model complex and non-linear relationships. There is a vast range of neural networks with diverse architectures. Readers are directed to literature on Recurrent Neural Networks (RNNs) [121], Convolutional Neural Networks (CNNs), and Graph Neural Networks (GNNs) for chemistry problems as they have had notable success, especially with language-based models (e.g., SMILES input) and molecular graph inputs [93,94]. Molecular graph inputs have the potential to highlight atoms or bonds in the molecule which were important for



**Fig. 8.** Illustration of a single layer perceptron.



**Fig. 9.** An exemplar neural network using the values for the Crabtree catalyst ‘toy problem’ (Scheme 1), where the presence of Ir and Rh could be encoded as “1” for present and “0” for absent, with a single hidden layer of eight nodes. The network is fully connected with a weight for each connection, but only one weight,  $w$ , is shown on the scheme for clarity. Note there is also an activation function between each layer and a bias between the hidden and output layer. The prediction is normalised to get the final activity of 0 or 1. After training, values for  $w$ , each node, and bias are established. For a new catalyst, e.g.,  $[\text{Ir}(\text{L}_\text{B})(\text{L}_\text{C}(\text{cod}))^+]$  as an input, with the respective descriptor values, the neural network could predict whether or not this catalysis is active.

the prediction. This can be used to verify the model is learning from the correct part of the molecule and to give insight into important structural features in the molecule [122]. We also refer readers to the following studies using neural networks in homogeneous catalysis prediction [123,124].

### 3.5. Toolkits and useful resources

Table 4 summarises some useful resources for ML, including some that are designed for the researcher with limited ML experience. We do not comment further here as extensive documentation and resources for these software sources are already widely available.

## 4. Case studies

### 4.1. Prediction of enantioselectivity in asymmetric catalysis using multiple linear regression

Binaphthylphosphoric acids, such as  $\text{L}_\text{C}\text{-H}$ , are strong activators of imines (**8**). Once protonated by acid, the resultant electrophilic iminiums (**9**) are readily attacked by a variety of nucleophiles ( $\text{Nu-H}$  = alcohols, thiols, phosphites, diazomethylphosphonates, diazoacetamides, hyperperoxides, Hantzsch esters, benzothiazolines and enecarbamates) the stereochemical outcome of which (*Re* or *Si* face addition of  $\text{Nu-H}$ ) is controlled by  $\text{L}_\text{C}$ . Reid and Sigman [101] postulated that known literature results for families of these additions could be parameterised allowing the performance of future unknown systems to be predicted. A dataset of 367 literature reactions was compiled. Categorising this by imine (*E*) or (*Z*) geometry allows insight into transition states associated with **9** as they give opposite absolute stereochemistry in the product amines. The authors arbitrarily assigned positive %ee values to reactions starting from (*E*)-geometry, and negative for (*Z*)-imines across all examples used to ensure consistent behaviour.

Molecular descriptors derived from DFT calculations were collected to describe the nucleophiles and catalysts that contained shared structural features. These DFT derived descriptors included: frontier orbital

**Table 4**

Selected commercial and open source resources for ML.

Tool Name	Description	Link
ChemDraw	Molecule drawing and labelling software (can be used for conversion into the SMILES format)	<a href="https://perkinelmerinformatics.com/products/research/chemdraw">https://perkinelmerinformatics.com/products/research/chemdraw</a>
Dragon	A commonly used application for the calculation of over 5000 molecular descriptors	<a href="https://chm.kode-solutions.net/pf/dragon-7-0/">https://chm.kode-solutions.net/pf/dragon-7-0/</a>
Google Colab	A free Jupyter notebook that runs entirely in the Google cloud, providing computational power for the development of deep learning and ML applications	<a href="https://colab.research.google.com/">https://colab.research.google.com/</a>
Microsoft Azure	Cloud computing service that includes Azure Cognitive Services for building intelligent applications based on AI/ML techniques	<a href="https://azure.microsoft.com/">https://azure.microsoft.com/</a>
Jupyter Notebook	Open source web-based environment for interactive computing providing services from data visualisation to ML and statistical modelling	<a href="https://jupyter.org/">https://jupyter.org/</a>
Orange	Open source ML and data visualisation software that is user-friendly and assumes little programming knowledge	<a href="https://orangedatamining.com/">https://orangedatamining.com/</a>
Scikit-learn	Open source Python library for machine learning	<a href="https://scikit-learn.org/stable/">https://scikit-learn.org/stable/</a>
PyTorch	Open source Python-based ML/deep learning library	<a href="https://pytorch.org/">https://pytorch.org/</a>
TensorFlow	Open source software for building ML pipelines and deep learning solutions	<a href="https://www.tensorflow.org/">https://www.tensorflow.org/</a>
WEKA	Open source ML toolbox accessible via a graphical user interface, standard terminal applications, or a Java API	<a href="https://www.weka.io/">https://www.weka.io/</a>
DeepChem	Open source Python library for machine learning and deep learning on molecular and quantum datasets	<a href="https://deepchem.io/">https://deepchem.io/</a>
MoleculeNet	A benchmark for testing and comparing molecular ML currently containing over 700,000 compounds	<a href="https://moleculenet.org/">https://moleculenet.org/</a>

energies (HOMO and LUMO) and important bond angles and bond lengths amongst others. Whilst the nucleophiles and catalysts in the training set have many common structural elements, the solvents used vary significantly in the data set. This needed additional parameterisation as the solvent used significantly affects the enantioselectivity of these reactions. In order to overcome this problem, a variety of 2D solvent descriptors were used including: molecular shape representation, size and number of heteroatoms present. Finally, categorical descriptors were added including reagent concentrations.

First, the key descriptors critical for highly enantioselective reactions were found. Linear regression was used to detect correlations between the experimentally observed enantioselectivity and combinations of the descriptors identified above. Cross-validation techniques (leave-one-out and *k*-fold) and external validation led to an acceptable linear regression model ( $R^2 = 0.88$ ). The resultant regression equation indicated that the main descriptor contributors are associated with the imine and the nucleophile, while the solvent and catalyst have a much less significant impact (Eq. 14). The equation coefficients are normalised to demonstrate the relative importance of each descriptor. This exemplifies an advantage of multiple linear regression and adds interpretability to the model as the magnitude and sign of each descriptor in the equation can be used to give insight into the importance and effect of the descriptors. While it would have been interesting to see if more complex ML methods

could build a better model; the small size of dataset may have precluded their use.

$$\Delta\Delta G^\ddagger = 0.42 + 0.29_{\text{sol}} - 0.90_{\text{NBO}_\text{N}} - 0.75_{\text{NBO}_\text{C}} + 0.33_{\text{L}_\text{s}} + 0.63_{\text{H-X-CNU}} + 0.20_{\text{L}_{\text{cat}}} \quad (14)$$

In Eq. 14: sol = solvent term (i.e. describing the positive impact of using aromatic solvent), NBO<sub>N</sub> and NBO<sub>C</sub> = imine natural bond orbital descriptors [which capture features of the imine that determine its transition state type pathway, different of (*E*) and (*Z*) imines], L<sub>s</sub> = a steric descriptor of smallest imine, H-X-CNU = a discovered critical bond angle, L<sub>cat</sub> = length of catalyst substituent on Ar group (i.e., a measure of its size).

In order to develop the simple model shown in Eq. 14 further, the dataset was split into (*E*)- and (*Z*)-imine derived transition states. A similar linear regression workflow was used (correlation between the experimentally observed %ee values and the now geometrically specific molecular structure descriptors). This analysis implied that for (*E*)-imines an energy minimising conformation that avoids repulsive interactions with large catalyst substituents is attained. Higher enantioselectivity is associated with large imine and catalyst substituents. However, for (*Z*)-imines enantiofacial selectivity the model was dominated by the nucleophile steric descriptor (larger nucleophiles correlate to higher product %ee).

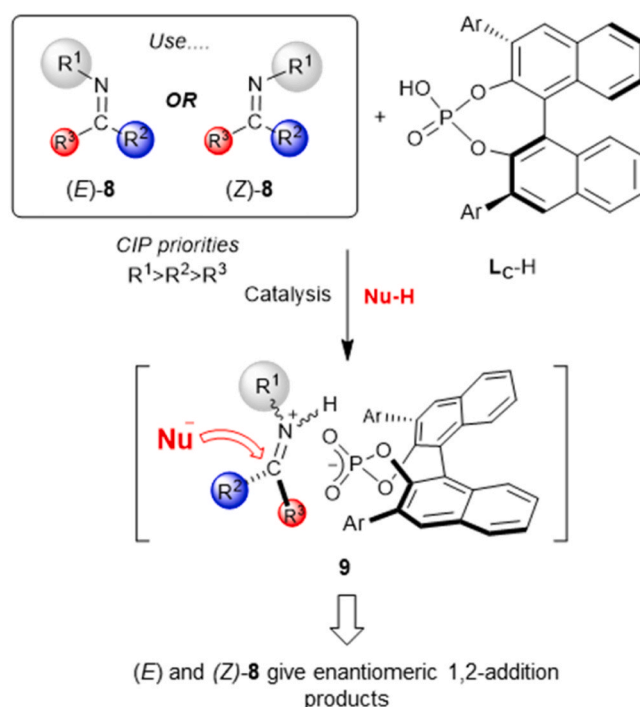
The validity of the transition state models was tested using motifs not included in the original training set. Three approaches were used. Firstly, the initial [fit to both (*E*) and (*Z*) data] model was tested using a nucleophile not included in the initial training. Of the 15 new reactions sampled, 13 were predicted within 5% of the observed %ee. Using an (*E*)-imine model alone, all reactions were predicted to the same  $\pm 5\%$  error bar. Secondly, the (*Z*)-imine model was tested on both a nucleophile and catalyst not contained in the original dataset. While 13 reactions were predicted a correct %ee value ( $\pm 2\%$ ), the model performed poorly in predicting the absolute stereochemistry.

In a final test of their approach, they focussed on the addition of thiols to benzoyl imines (following the work of Zahrt et al. [125]). All reactions of this type were removed, and the model was retrained accordingly. The initial [(*E*) and (*Z*) data] model still predicted the enantioselectivity (%ee) of 26 out of 34 reactions within 5%. In the case of (*E*)-imines the correct absolute stereochemistry was predicted in 25 cases. The advantage of the Reid and Sigman approach is that using multiple linear regression associated to a range of 'simple' models, allows the data to be more easily interpreted.

#### 4.2. Predictions of higher-selectivity catalysts for chiral phosphoric acid-catalyzed thiol additions

Zahrt et al. [125] also adopted a data-driven process for the development of novel catalysts for chiral phosphoric acid-catalyzed thiol additions to *N*-acylimines in a similar manner to Scheme 4. Their workflow comprised: development of an *in silico* library, calculation of descriptors, selection of a representative subset of the catalysts, training data, and use of SVM, random forest and neural network methods to predict reaction enantioselectivity (%ee). Most significantly a new average steric occupancy (ASO) descriptor was developed. ASO simplifies the number of conformers into a numerical form based on location by conflating conformers if they are within a defined van der Waals radius allowing improved computational efficiency. Electronic descriptors were generated through calculated interactions with a quaternary ammonium ion. This study demonstrated the importance of descriptor selection as they had a very large number of descriptors (16, 384). This was reduced by removing any descriptors with a variance of zero since they do not add any information to the model. Then PCA was used to reduce the dimensionality of the descriptor space.

ML training was initiated on a previously optimised reaction (enantioselective formation of *N,S*-acetals), with a training set of 600



Scheme 4. Asymmetric 1,2-additions to geometrical isomers of imines 8.

and a test set of 475 reactions. Overall, an SVM approach was found to be the most effective method for catalyst selectivity prediction for randomly selected train and test sets. It is possible SVM was the best machine learning method at finding trends in the large input space used in this study, though it should be noted that often multiple machine learning approaches should be compared to get the most accurate model. Catalyst performance was measured by comparing predicted vs. observed  $\Delta\Delta G^\ddagger$  (kcal mol<sup>-1</sup>) values. The model predicted enantioselectivity within 0.3–0.4 kcal mol<sup>-1</sup>. The most accurately modelled catalyst had predicted %ee within 1–2% of the experimental %ee.

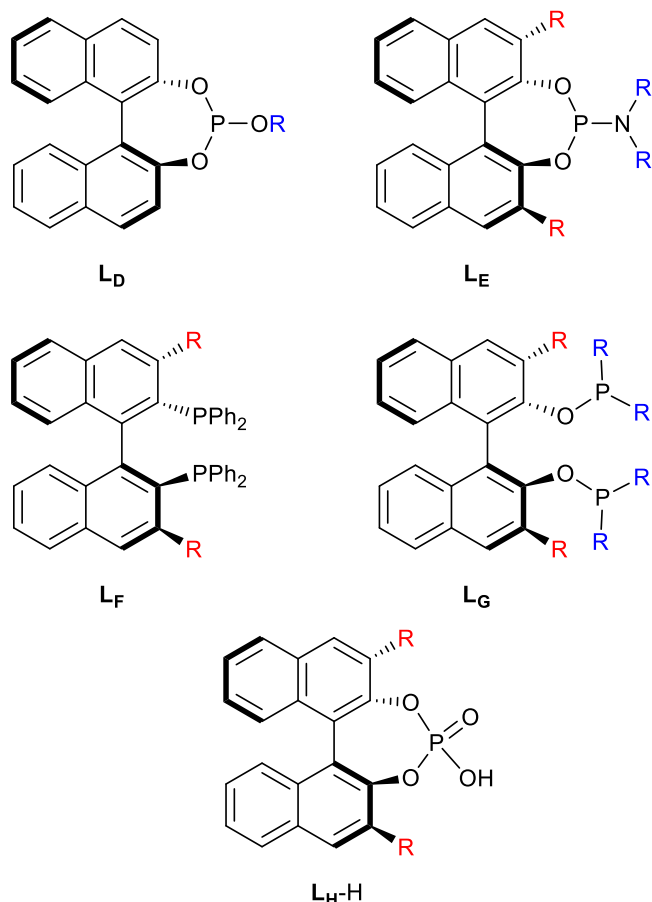
#### 4.3. A unified ML protocol for asymmetric catalysis as a proof of concept demonstration using asymmetric hydrogenation

Singh et al. [126] applied random forest to predict the outcome of asymmetric hydrogenation. An *in silico* library of 386 asymmetric hydrogenation reactions was built, with five catalyst families based on an (*S*) axially chiral binaphthyl ligands. The five families chosen were: 1. mono-BINOL-phosphites (L<sub>D</sub>), 2. BINOL-phosphoramidites (L<sub>E</sub>), 3. BINAP derivatives (L<sub>F</sub>), 4. bis-BINOL-phosphites (L<sub>G</sub>) and 5. BINOL-phosphoric acids (L<sub>H-H</sub>) (Scheme 5).

They selected a broad number of descriptors for each substrate and catalyst. The catalysts were typically ruthenium(II) or rhodium(I) complexes of the ligands of Scheme 5. As the catalysis is ligand controlled, 22 global descriptors represented overall molecular properties of the general ligand, e.g., HOMO, LUMO, polar surface area and dipole moment. Eight site-specific descriptors represented differences arising from substituents. These descriptors included: bond length and angles, vibrational frequencies and intensities, NMR chemical shifts, atomic charges, and distance between nonbonded atoms.

The ML model was developed to predict the hydrogenation product %ee value as a function of ligand. A random forest model was built using 20% of the instances for the test set and 80% of instances for the training set. To ensure that the model was sufficiently rigorous, 100 test-train sets were used, each produced by random selection and five-fold cross validation. These test-train sets were performed to identify the best "hyper-parameter selection". Using these hyper-parameters the average





**Scheme 5.** Ligand families used in the study of Singh et al.; R represent points of ligand variation within the families  $L_D$ - $L_H$ .

performance of the 100 test-train sets was measured and then compared with experimental data of the test set. For each of the five catalyst families an independent random forest model was constructed. For example, in the mono-BINOL-phosphite catalyst family ( $L_D$ ) only relevant reactions were incorporated into the test train sets. The outcome of this was five models with an RMSE of 5.4% for the %ee and a demonstration of the model's ability to decipher the relationships between the descriptors, the enantioselectivity and the substrate-catalyst combinations.

A unified random forest model was created by first combining the data sets from families 1 ( $L_D$ ) and 2 ( $L_E$ ) and training a random forest model on this set. The predicted %ee had an RMSE of 8.5% with respect to the experimental %ee. This was further developed by adding the data set for the fifth catalyst family ( $L_H$ -H); which had an RMSE of 6.8%. In the final model all 368 reactions were used and showed an RMSE of 8.4%. This model was also tested on varying sizes of data set with RMSEs less than 10 for data sets from 39 reactions to 386 reactions. The random forest model was compared with other commonly used ML techniques such as convolutional neural networks and extreme gradient boosting which showed RMSEs of 9.6% and 11.6% compared to the random forest RMSE of 9.2%. It is likely that the relatively small size of the dataset made random forest a better choice than a neural network. Additionally, random forest made the subsequent analysis of descriptor important easier, adding interpretability to the model. The final part of this work was to identify chemical patterns using decision trees with the rationale of discovering how variations in the molecular descriptors can be used to fine tune %ee. This was achieved by having 20% of the data set as holdout (data excluded from the model on which to test the model) in each run. In total 100 runs were performed where in each one a critical

descriptor was varied and at the end the best decision tree was selected. In this particular case vibrational intensities, dihedral angles, bond angles, dipole moments and volume were identified as key descriptors. This information is potentially useful in designing later ML-improved ligand candidates, a concept explored in the next example.

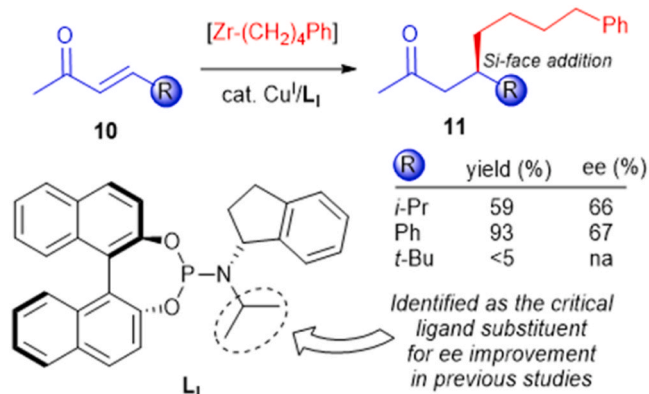
#### 4.4. Retooling asymmetric conjugate additions for sterically demanding substrates with an iterative data-driven approach

The asymmetric conjugate addition (ACA) of zirconium organometallics (specifically  $Cp_2ZrCl(CH_2)_4Ph$ ) to enones **10** generates the useful chiral ketones **11** [127]. This procedure had a limitation that branched substituents 'R' provided only low levels of enantioselectivity (Scheme 6, typically %ee values over 90% are required for onward synthetic use of **11**). Preliminary studies had already shown that the binaphthol and indane units of  $L_I$  were already close to optimal, but the effect of the isopropyl region of ligand space on the reaction %ee was not clear.

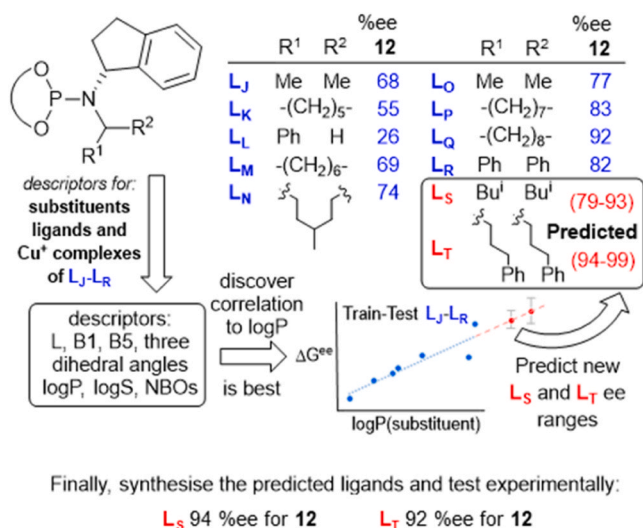
Nine ligands ( $L_J$ - $L_R$ ), differing only at the isopropyl substituent position, were synthesised and simultaneously descriptors for their substituents alone, the ligands themselves and their Cu(I) complexes were generated using DFT obtained geometries (Scheme 7).

Sub-sets of six ligands at a time were used to discover correlations of the difference in  $\Delta G^\ddagger$  (directly related to the %ee of **12**) to these descriptors (or combinations thereof). Three instructive facts are uncovered in this process. Firstly, regression fits of a single input are preferred, as with only six instances (ligands), overfitting of the data becomes a common problem. Secondly, due to limited instances regression fit of ensembles of conformational geometry and steric descriptors (ex. DFT/Sterimol) did not proceed well. However, acceptable ( $R^2_{test} > 0.6$ , RMSE ( $\Delta\Delta G^\ddagger_{test}$ )  $< 2$  kJmol $^{-1}$ ) forward correlation to the substituent lipophilicity (logP) descriptor was found. This allowed guided prediction that  $L_S$  and  $L_T$  (higher logP) would provide higher %ee in ACA reactions yielding **12**. Both  $L_S$  and  $L_T$  were human (as opposed to ML) designed and synthesised. A third important fact is that simply predicting the 'best' ligand in the absence of understanding the confidence/error in the regression analysis is a poor strategy. The regression of Scheme 7 suggests  $L_T$  to be the best ligand but is of significant uncertainty compared to the predicted  $L_S$  %ee value. In fact, their true experimental performance is actually reversed by a  $\Delta\Delta G^\ddagger_{obs}$  of  $\sim 1$  kJmol $^{-1}$  (94 vs. 92% ee for **12** using  $L_S$  compared to  $L_T$ ). Keeping track of the confidence bounds in ML predictions is, thus, very important.

The model of Scheme 7 was unable to provide insight into the exact ligand structural features required to maximise the %ee of **12**. For example, additional ligand candidates with high logP delivered slightly poorer experimental ee, while a simple CHET<sub>2</sub> substituent provided the same enantioselectivity as  $L_T$  in the lab, despite having a much lower calculated logP value. In order to address these deviations a significant number of additional ligand instances (22 synthesised and 20 'computed



**Scheme 6.** Problem substrates **10** for ACA reactions promoted by ligand  $L_I$ .



**Scheme 7.** Simple selection of an optimal ligand training descriptor for formation of 12.

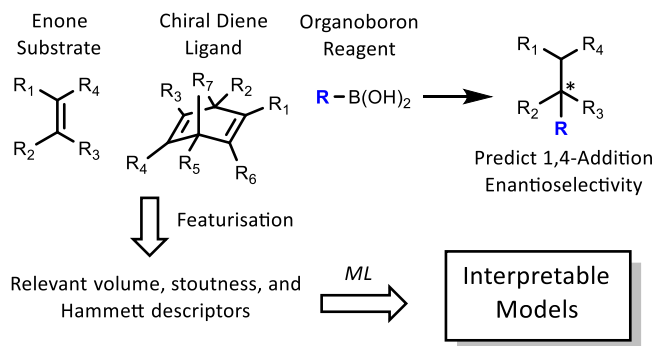
only' structures) were added to the data pool in a stepwise manner. This allowed improved regression analyses and an improved (higher confidence level) model using a combination of the ligand  $E_{\text{HOMO}}$  and minimum energy conformation dipole moment descriptors to be developed. However, this came at the cost of a significant increase in DFT conformational analysis power/time required. While no further gains in ACA % ee performance were attained, importantly this second model allowed the identification of a gauche conformation in CH(n-alkyl)<sub>2</sub> substituents as being critical to attaining high stereoselectivity. Finally, it also indicated that further ligand development of this ligand feature would not be profitable, as the system had 'topped out' at 94%ee.

#### 4.5. Machine learnt patterns in rhodium-catalysed asymmetric Michael addition using chiral diene ligands

Rhodium-catalysed asymmetric 1,4-addition of organoboron reagents to Michael acceptors is a frequently used methodology where high levels of enantioselectivity is often obtained [128]. The reaction comprises of an enone substrate (*i.e.*, an alkene with an electron-withdrawing group), a chiral diene ligand to direct the stereochemistry, and an organoboron reagent. Hayashi proposed a steric-dominated selectivity model [128]. However, this does not take into account electronic considerations, known to be important in the rate-determining step of this reaction. DFT models have also been built for this reaction, but this can be time-consuming if several transition states must be found. Thus, a simple supervised ML regression model was built to predict the '%top' product formed [129]. This metric was chosen instead of the typical R/S stereochemistry classification since R/S is based on R-group atoms and molecular weights. The metric '% top' was defined relative to the electron withdrawing group in the enone, making it suitable for ML. Scheme 8 shows the reaction and ML workflow.

This study highlighted the "black-box" problem in synthetic chemistry, where ML models are perceived as opaque and overly complicated. Therefore, an interpretable ML model was built which could enhance the chemist's understanding of the reaction. The enone, chiral diene ligand, and organoboron reagent were featurised into a small number of descriptors which were known to be important, according to the current understanding of the reaction. These features were volumes and Hammett constants of the R<sub>1</sub>-R<sub>7</sub> groups on the chiral diene ligand, volume of the organoboron residue, and stoutness (volume/longest chain length) of the substrate R<sub>1</sub>-R<sub>4</sub> groups.

From a collection of 4303 examples, 610 reactions were curated.



**Scheme 8.** Interpretable models were built from a small number of carefully chosen descriptors. These models gave insight into the enantioselectivity of 1,4-addition reactions.

These data were split into training and test data (in a 9:1 ratio) and 5-fold cross validation was run on the training data. A total of 30 random test-train splits were performed to get mean predictions with their standard deviations (as a measure of error).

ML models were built with Random Forest, Multiple Linear Regression, and Gradient Boosted Regression [130] (an ensemble method like Random Forest). All the ensemble methods gave comparable results and outperformed Multiple Linear Regression, showing the power of ensemble methods, even for a modestly sized dataset. The models gave a lowest RMSE of 9% and  $R^2$  of 0.7–0.8, comparable to models built using 11514 DRAGON descriptors or molecular signatures (fragment-based). The performance of the DRAGON model, in contrast to the main model in the paper, was significantly worse on the test data than training data. This indicated overfitting, most likely due to the large number of descriptors. Another advantage of the main model was high levels of interpretability. The importance of the descriptors was ranked and both steric and electronic descriptors were weighted highly. Additionally, this model was better than the DRAGON model at identifying outliers by analysing which examples were poorly predicted; these outliers were indicated by high standard deviation (variation between models built with different samples of the training data). Therefore, due to the interpretability of this model, this methodology could allow identification of promising *de novo* ligands for asymmetric organic synthesis.

#### 4.6. Machine learning directs design of Cr olefin oligomerisation catalysts

Short chain linear olefins, such as 1-hexene and 1-octene, are important in the manufacture of several important products such as plasticisers and lubricants. A study was devised to combine quantum mechanical transition states with machine learning to design new catalysts which could selectively produce 1-octene from ethylene [108]. Most interestingly, the primary aim of the study was not to predict selectivity but to determine which descriptors were most important and use those descriptors to design new catalysts.

The system under study was Cr and 105 unique phosphine imine *P,N* ligands. Transition state energies for the two different transition states to form 1-hexene and 1-octene were determined using DFT. The resultant structures were used to calculate 14 atomic and molecular descriptors. These included geometric features such as bond lengths and angles, percent volume buried, and distance out of pocket. Percent volume buried described the extent to which the first coordination sphere of the Cr metal centre is occupied by a *P,N* ligand. The distance out of pocket describes the how far the Cr metal is situated from the *P,N* ligand. Using these descriptors, several machine learning methods, including random forest and SVM, were used to predict the energy difference between the two competing transition states. They found that random forest gave the best models. Although the authors do not explain why random forest gave the best model, they note, as we do, the success and versatility of the method for small dataset homogeneous catalysis systems. Random

forest gave the lowest RMSE and correctly predicted the overall selectivity for 83 out of 105 ligands. They note the skew in their dataset as more of the ligands were 1-octene selectivity, and posit a more balanced dataset could have enhanced the predictions. The random forest model was interpreted by determining the descriptor importance. Three important descriptors were identified: Cr-N distance; Cr- $\alpha$  distance; and distance out of pocket. This led to the generation of seven new ligands by modifying ligands to change these important descriptor values. When experimentally verified, the newly designed catalysts gave > 95% 1-octene selectivity.

## 5. Conclusion

This overview aimed to demonstrate that *ML* algorithms are a valuable ally to the synthetic researcher, and more accessible than at first appearance. We hope the contents of this article will prompt further collaboration between the chemical and computational sciences in furthering the field of *ML* in chemistry. Under the umbrella of “Machine Learning in Ligand Promoted Catalysis” work that has been achieved by researchers includes: rapid virtual catalyst screening; significant improvements molecular descriptors; development of successful novel ligands (and enzymes); enhanced mechanistic understanding or elucidation; improved enantioinduction; generally focussed on a modest ‘area’ of catalytic reaction space; prediction of reaction or reactant types not supplied to the algorithm in the training set; and acceleration of traditional empirical catalyst discovery methods.

Areas where we expect further development in the near or more distant future include: greater need for consistent format in published *ML* literature and consistent use of nomenclature across the catalysis-informatics divide; text- and image-mining for reaction and catalyst data stored in text, tables and schemes; faster quantum chemistry calculations to guide predictive models; need for the practicing chemist to have basic knowledge of *ML* methods; increased publication of ‘failed’ experiments to decrease database bias; alternative ways to model the surface of the catalytic pocket; identification of species in microscopic and spectroscopic analysis by *ML*; and increasing use of explainable and interpretable AI, to provide models which chemists trust and add to the knowledge of catalytic systems.

As the popularity of *ML* increases, coupled with the improvements in computational power, the potential in chemistry could result in a whole new way of experimentation and add another tool to the chemists’ toolkit. However, one thing is clear: chemists should be encouraged to engage with the process of developing AI and *ML* in chemistry, and in educating the next generation of scientists as the field evolves.

## Funding Information

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Sustainable Chemistry (grant number EP/S022236/1). Prof Hirst is supported by the Royal Academy of Engineering under the Chairs in Emerging Technologies scheme.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] K.T. Butler, D.W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science, *Nature* 559 (2018) 547–555.
- [2] M.H.S. Segler, M. Preuss, M.P. Waller, Planning chemical syntheses with deep neural networks and symbolic AI, *Nature* 555 (2018) 604–610.
- [3] H. Gao, T.J. Struble, C.W. Coley, Y. Wang, W.H. Green, K.F. Jensen, Using machine learning to predict suitable conditions for organic reactions, *ACS Cent. Sci.* 4 (2018) 1465–1476.
- [4] A.G. Maldonado, G. Rothenberg, Predictive modeling in homogeneous catalysis: a tutorial, *Chem. Soc. Rev.* 39 (2010) 1891–1902.
- [5] A.F. Zahrt, Sv Athavale, S.E. Denmark, Quantitative structure–selectivity relationships in enantioselective catalysis: Past, present, and future, *Chem. Rev.* 120 (2019) 1620–1689.
- [6] S. Palkovits, A primer about Machine Learning in Catalysis—A Tutorial with Code, *ChemCatChem* 12 (2020) 3995–4008.
- [7] F. Strieth-Kalthoff, F. Sandfort, M.H.S. Segler, F. Glorius, Machine learning the ropes: principles, applications and directions in synthetic chemistry, *Chem. Soc. Rev.* 49 (2020) 6154–6168.
- [8] G. dos Passos Gomes, R. Pollice, A. Aspuru-Guzik, Navigating through the maze of homogeneous catalyst design with machine learning, *Trends Chem.* 3 (2021) 96–110.
- [9] W. Yang, T.T. Fidelis, W.H. Sun, Machine learning in catalysis, from proposal to practicing, *ACS Omega* 5 (2020) 83–88.
- [10] P. SchlexerLamoureux, K.T. Winther, J.A. GarridoTorres, V. Streibel, M. Zhao, M. Bajdich, F. Abild-Pedersen, T. Bligaard, Machine learning for computational heterogeneous catalysis, *ChemCatChem* 11 (2019) 3581–3601.
- [11] S. Singh, R.B. Sunoj, Molecular machine learning for chemical catalysis: prospects and challenges, *Acc. Chem. Res.* 56 (2023) 402–412.
- [12] D.J. Gorin, B.D. Sherry, F.D. Toste, Ligand effects in homogeneous Au catalysis, *Chem. Rev.* 108 (2008) 3351–3378.
- [13] B. Cornils, W.A. Herrmann, Concepts in homogeneous catalysis: the industrial view, *J. Catal.* 216 (2003) 23–31.
- [14] L. Kollár, G. Keglevich, P-heterocycles as ligands in homogeneous catalytic reactions, *Chem. Rev.* 110 (2010) 4257–4302.
- [15] M. Renom-Carrasco, L. Lefort, Ligand libraries for high throughput screening of homogeneous catalysts, *Chem. Soc. Rev.* 47 (2018) 5038–5060.
- [16] R. Leardi, Experimental design in chemistry: a tutorial, *Anal. Chim. Acta* 652 (2009) 161–172.
- [17] K.M. Jablonka, L. Patiny, B. Smit, Making the collective knowledge of chemistry open and machine actionable, *Nat. Chem.* 14 (2022) 365–376.
- [18] S. Sun, M. Niranjana, S. Kanza, J.G. Frey, A review of reinforcement learning in chemistry, *Digital Discovery* 1 (2022) 551–567.
- [19] A. Glielmo, B.E. Husic, A. Rodriguez, C. Clementi, F. Noé, A. Laio, Unsupervised learning methods for molecular simulation data, *Chem. Rev.* 121 (2021) 9722–9758.
- [20] J. Polanski, Unsupervised learning in drug design from self-organization to deep chemistry, *Int. J. Mol. Sci.* 23 (2022) 2797.
- [21] S. Sun, Z. Cao, H. Zhu, J. Zhao, A survey of optimization methods from a machine learning perspective, *IEEE Trans. Cybern.* 50 (2019) 3668–3681.
- [22] H. Song, I. Triguero, E. Özcan, A review on the self and dual interactions between machine learning and optimisation, *Prog. Artif. Intell.* 8 (2019) 143–165.
- [23] R. Crabtree, Iridium compounds in catalysis, *Acc. Chem. Res.* 12 (1979) 331–337.
- [24] L. Massaro, J. Zheng, C. Margarita, P.G. Andersson, Enantioconvergent and enantiodivergent catalytic hydrogenation of isomeric olefins, *Chem. Soc. Rev.* 49 (2020) 2504–2522.
- [25] S.H. Shetty, S. Shetty, C. Singh, A. Rao, Supervised machine learning: algorithms and applications, in: P. Singh (Ed.), *Fundamentals and Methods of Machine and Deep Learning: Algorithms, Tools and Applications*, John Wiley & Sons, 2022, pp. 1–16.
- [26] S.B. Kotsiantis, I. Zaharakis, P. Pintelas, Supervised machine learning: a review of classification techniques, *Emerg. Artif. Intell. Appl. Comput. Eng.* 160 (2007) 3–24.
- [27] I.V. Tetko, O. Engkvist, U. Koch, J.L. Reymond, H. Chen, BIGCHEM: challenges and opportunities for big data analysis in chemistry, *Mol. Inf.* 35 (2016) 615–621.
- [28] J.M. Cole, The chemistry of errors, *Nat. Chem.* 14 (2022) 973–975.
- [29] S.M. Kearnes, M.R. Maser, M. Wlekliński, A. Kast, A.G. Doyle, S.D. Dreher, J. M. Hawkins, K.F. Jensen, C.W. Coley, The open reaction database, *J. Am. Chem. Soc.* 143 (2021) 18820–18826.
- [30] P. Baldi, Call for a public open database of all chemical reactions, *J. Chem. Inf. Model* 62 (2022) 2011–2014.
- [31] Y. Haraguchi, Y. Igarashi, H. Imai, Y. Oaki, Sparse modeling for small data: case studies in controlled synthesis of 2D materials, *Digit. Discov.* 1 (2022) 26–34.
- [32] J.A. Esterhuizen, B.R. Goldsmith, S. Linic, Interpretable machine learning for knowledge generation in heterogeneous catalysis, *Nat. Catal.* 5 (2022) 175–184.
- [33] F. Oviedo, J.L. Ferres, T. Buonassisi, K.T. Butler, Interpretable and explainable machine learning for materials science and chemistry, *Acc. Mater. Res.* 3 (2022) 597–607.
- [34] J. Jiménez-Luna, F. Grisoni, G. Schneider, Drug discovery with explainable artificial intelligence, *Nat. Mach. Intell.* 2 (2020) 573–584.
- [35] M. Haghighatli, J. Li, F. Heidar-Zadeh, Y. Liu, X. Guan, T. Head-Gordon, Learning to make chemical predictions: the interplay of feature representation, data, and machine learning methods, *Chem* 6 (2020) 1527–1542.
- [36] N. Artrith, K.T. Butler, F.X. Coudert, S. Han, O. Isayev, A. Jain, A. Walsh, Best practices in machine learning for chemistry, *Nat. Chem.* 13 (2021) 505–508.
- [37] D.T. Ahneman, J.G. Estrada, S. Lin, S.D. Dreher, A.G. Doyle, Predicting reaction performance in C–N cross-coupling using machine learning, *Science* 360 (2018) 186–190.
- [38] J.B.O. Mitchell, Machine learning methods in chemoinformatics, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 4 (2014) 468–481.
- [39] B. Krawczyk, Learning from imbalanced data: open challenges and future directions, *Prog. Artif. Intell.* 5 (2016) 221–232.



- [40] A. Famili, W.M. Shen, R. Weber, E. Simoudis, Data preprocessing and intelligent data analysis, *Intell. Data Anal.* 1 (1997) 3–23.
- [41] B. Efron, Missing data, imputation, and the bootstrap, *J. Am. Stat. Assoc.* 89 (1994) 463–475.
- [42] T.M. Whitehead, B.W.J. Irwin, P. Hunt, M.D. Segall, G.J. Conduit, Imputation of assay bioactivity data using deep learning, *J. Chem. Inf. Model* 59 (2019) 1197–1204.
- [43] M.M. Ahsan, M.A.P. Mahmud, P.K. Saha, K.D. Gupta, Z. Siddique, Effect of data scaling methods on machine learning algorithms and model performance, *Technologies* 9 (2021) 52.
- [44] H. He, E.A. Garcia, Learning from imbalanced data, *IEEE Trans. Knowl. Data Eng.* 21 (2009) 1263–1284.
- [45] J. Brownlee, *Machine learning algorithms from scratch with python*, Machine Learning Mastery, 2016.
- [46] P. Branco, L. Torgo, R.P. Ribeiro, Pre-processing approaches for imbalanced distributions in regression, *Neurocomputing* 343 (2019) 76–99.
- [47] D.M. Hawkins, The problem of overfitting, *J. Chem. Inf. Comput. Sci.* 44 (2004) 1–12.
- [48] J. Weng, D.S. Young, Some dimension reduction strategies for the analysis of survey data, *J. Big Data* 4 (2017) 1–19.
- [49] J. Miao, L. Niu, A survey on feature selection, *Procedia Comput. Sci.* 91 (2016) 919–926.
- [50] I. Ponzoni, V. Sebastián-Pérez, C. Requena-Triguero, C. Roca, M.J. Martínez, F. Cravero, M.F. Díaz, J.A. Páez, R.G. Arrayás, J. Adrio, N.E. Campillo, Hybridizing feature selection and feature learning approaches in QSAR modeling for drug discovery, *Sci. Rep.* 7 (2017) 1–19.
- [51] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, M. Lang, Benchmark for filter methods for feature selection in high-dimensional classification data, *Comput. Stat. Data Anal.* 143 (2020), 106839.
- [52] M. Eklund, U. Norinder, S. Boyer, L. Carlsson, Choosing feature selection and learning algorithms in QSAR, *J. Chem. Inf. Model.* 54 (2014) 837–843.
- [53] S.A. Alsenan, I.M. Al-Turaiki, A.M. Hafez, Feature extraction methods in quantitative structure–activity relationship modeling: a comparative study, *IEEE Access* 8 (2020) 78737–78752.
- [54] M.G. Gustafsson, Independent component analysis yields chemically interpretable latent variables in multivariate regression, *J. Chem. Inf. Model.* 45 (2005) 1244–1255.
- [55] A. Giuliani, The application of principal component analysis to drug discovery and biomedical data, *Drug Discov. Today* 22 (2017) 1069–1076.
- [56] H. Abdi, L.J. Williams, Principal component analysis, *Wiley Interdiscip. Rev. Comput. Stat.* 2 (2010) 433–459.
- [57] W.J. Dunn Iii, S. Wold, U. Edlund, S. Hellberg, J. Gasteiger, Multivariate structure–activity relationships between data from a battery of biological tests and an ensemble of structure descriptors: the PLS method, *Quant. Struct. Relatsh.* 3 (1984) 131–137.
- [58] R.D. Cramer, Partial least squares (PLS): its strengths and limitations, *Perspect. Drug Discov. Des.* 1 (1993) 269–278.
- [59] T. Mehmood, B. Ahmed, The diversity in the applications of partial least squares: an overview, *J. Chemom.* 30 (2016) 4–17.
- [60] L. McInnes, J. Healy, J. Melville, UMAP: Unif. Manifold Approx. Proj. Dimens. Reduct. 2020 Available from doi: 10.48550/arXiv.1802.03426.
- [61] M.C. Sorkun, D. Mullaj, J.M.V.A. Koelman, S. Er, ChemPlot, a python library for chemical space visualization, *Chem. Methods* 2 (2022), e202200005.
- [62] D. Berrar, Cross-validation, in: S. Ranganathan, M. Gribskov, K. Nakai, C. Schönbach (Eds.), *Encyclopedia of Bioinformatics and Computational Biology*, 2019, pp. 542–545.
- [63] G.C. Cawley, N.L.C. Talbot, On over-fitting in model selection and subsequent selection bias in performance evaluation, *J. Mach. Learn. Res.* 11 (2010) 2079–2107.
- [64] H. Hong, Q. Xie, W. Ge, F. Qian, H. Fang, L. Shi, Z. Su, R. Perkins, W. Tong, Mold2, molecular descriptors from 2D structures for chemoinformatics and toxicoinformatics, *J. Chem. Inf. Model.* 48 (2008) 1337–1344.
- [65] L. Wang, J. Ding, L. Pan, D. Cao, H. Jiang, X. Ding, Quantum chemical descriptors in quantitative structure–activity relationship models and their applications, *Chemom. Intell. Lab. Syst.* 217 (2021), 104384.
- [66] S. Mapari, S.K.V. Camarda, Use of three-dimensional descriptors in molecular design for biologically active compounds, *Curr. Opin. Chem. Eng.* 27 (2020) 60–64.
- [67] R.D. Cramer, D.E. Patterson, J.D. Bunce, Comparative molecular field analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins, *J. Am. Chem. Soc.* 110 (1988) 5959–5967.
- [68] J.L. Melville, K.R.J. Lovelock, C. Wilson, B. Allbutt, E.K. Burke, B. Lygo, J. D. Hirst, Exploring phase-transfer catalysis with molecular dynamics and 3D/4D quantitative structure–selectivity relationships, *J. Chem. Inf. Model* 45 (2005) 971–981.
- [69] C.L. Senese, J. Duca, D. Pan, A.J. Hopfinger, Y.J. Tseng, 4D-fingerprints, universal QSAR and QSPR descriptors, *J. Chem. Inf. Comput. Sci.* 44 (2004) 1526–1539.
- [70] D. Fourches, J. Ash, 4D-quantitative structure–activity relationship modeling: making a comeback, *Expert Opin. Drug Discov.* 14 (2019) 1227–1235.
- [71] C. Hansch, A. Leo, R.W. Taft, A survey of Hammett substituent constants and resonance and field parameters, *Chem. Rev.* 91 (1991) 165–195.
- [72] P. Ertl, A. Web, Tool for calculating substituent descriptors compatible with hammett sigma constants, *Chem. Methods* 2 (2022), e202200041.
- [73] C.A. Tolman, Phosphorus ligand exchange equilibria on zerovalent nickel. Dominant role for steric effects, *J. Am. Chem. Soc.* 92 (1970) 956–2965.
- [74] J. Jover, J. Cirera, Computational assessment on the Tolman cone angles for P-ligands, *Dalton Trans.* 48 (2019) 15036–15048.
- [75] N. Govindarajan, H. Beks, E.J. Meijer, Variability of ligand pKa during homogeneously catalyzed aqueous methanol dehydrogenation, *ACS Catal.* 10 (2020) 14775–14781.
- [76] M.D. Wodrich, B. Sawatlon, E. Solel, S. Kozuch, C. Corminboeuf, Activity-based screening of homogeneous catalysts through the rapid assessment of theoretically derived turnover frequencies, *ACS Catal.* 9 (2019) 5716–5725.
- [77] M. Haghighatlari, J. Li, F. Heidar-Zadeh, Y. Liu, X. Guan, T. Head-Gordon, Learning to make chemical predictions: the interplay of feature representation, data, and machine learning methods, *Chem* 6 (2020) 1527–1542.
- [78] S. Wang, J. Jiang, Interpretable catalysis models using machine learning with spectroscopic descriptors, *ACS Catal.* 13 (2023) 7428–7436.
- [79] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, *J. Chem. Inf. Comput. Sci.* 28 (1988) 31–36.
- [80] N.M. O’Boyle, M. Banck, C.A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, Open babel: an open chemical toolbox, *J. Chemin.* 3 (2011) 1–14.
- [81] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi, I. Pletnev, InChI-the worldwide chemical structure identifier standard, *J. Chemin.-.* 5 (2013) 1–9.
- [82] I. Pletnev, A. Erin, A. McNaught, K. Blinov, D. Tchekhovskoi, S. Heller, InChIKey collision resistance: an experimental testing, *J. Chemin.-.* 4 (2012) 1–9.
- [83] M. Sastry, J.F. Lowrie, S.L. Dixon, W. Sherman, Large-scale systematic analysis of 2D fingerprint methods and parameters to improve virtual screening enrichments, *J. Chem. Inf. Model.* 50 (2010) 771–784.
- [84] I. Muegge, P. Mukherjee, An overview of molecular fingerprint similarity search in virtual screening, *Expert Opin. Drug Discov.* 11 (2016) 137–148.
- [85] J.L. Durant, B.A. Leland, D.R. Henry, J.G. Nourse, Reoptimization of MDL keys for use in drug discovery, *J. Chem. Inf. Comput. Sci.* 42 (2002) 1273–1280.
- [86] PubChem Database, <http://pubchem.ncbi.nlm.nih.gov> (accessed May 2023).
- [87] Daylight chemical information systems, Daylight, <http://www.daylight.com/>, (accessed May 2023).
- [88] D. Rogers, M. Hahn, Extended-connectivity fingerprints, *J. Chem. Inf. Model* 50 (2010) 742–754.
- [89] M.J. McGregor, S.M. Muskal, Pharmacophore fingerprinting. 1. Application to QSAR and focused library design, *J. Chem. Inf. Comput. Sci.* 39 (1999) 569–574.
- [90] T. Imamoto, Searching for practically useful P-chirogenic phosphine ligands, *Chem. Rec.* 16 (2016) 2659–2673.
- [91] J. Hua, Z. Xiong, J. Lowey, E. Suh, E.R. Dougherty, Optimal number of features as a function of sample size for various classification rules, *Bioinformatics* 21 (2005) 1509–1515.
- [92] C. Beileites, U. Neugebauer, T. Bocklitz, C. Krafft, J. Popp, Sample size planning for classification models, *Anal. Chim. Acta* 760 (2013) 25–33.
- [93] C.W. Coley, W. Jin, L. Rogers, T.F. Jamison, T.S. Jaakkola, W.H. Green, R. Barzilay, K.F. Jensen, A graph-convolutional neural network model for the prediction of chemical reactivity, *Chem. Sci.* 10 (2019) 370–377.
- [94] Z. Tu, C.W. Coley, Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction, *J. Chem. Inf. Model.* 62 (2022) 3503–3513.
- [95] R. Parikh, A. Mathai, S. Parikh, G.C. Sekhar, R. Thomas, Understanding and using sensitivity, specificity and predictive values, *Indian J. Ophthalmol.* 56 (2008) 45.
- [96] D.S. Palmer, N.M. O’Boyle, R.C. Glen, J.B.O. Mitchell, Random forest models to predict aqueous solubility, *J. Chem. Inf. Model.* 47 (2007) 150–158.
- [97] A. Kaur, R. Kumar, Comparative analysis of parametric and non-parametric tests, *J. Comput. Math. Sci.* 6 (2015) 336–342.
- [98] E. González-Estrada, W. Cosmes, Shapiro–Wilk test for skew normal distributions based on data transformations, *J. Stat. Comput. Simul.* 89 (2019) 3258–3272.
- [99] J. Demsar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (2006) 1–30.
- [100] D.J. Olive, Multiple linear regression, in: *Linear Regression*, Springer, 2017, pp. 17–83.
- [101] J.P. Reid, M.S. Sigman, Holistic prediction of enantioselectivity in asymmetric catalysis, *Nature* 571 (2019) 343–348.
- [102] C. Kingsford, S.L. Salzberg, What are decision trees? *Nat. Biotechnol.* 26 (2008) 1011–1013.
- [103] G. Biau, E. Scornet, A random forest guided tour, *Test* 25 (2016) 197–227.
- [104] N. Altman, M. Krzywinski, Ensemble methods: bagging and random forests, *Nat. Methods* 14 (2017) 933–935.
- [105] F. Tang, H. Ishwaran, Random forest missing data algorithms, *Stat. Anal. Data Min.: ASA Data Sci. J.* 10 (2017) 363–377.
- [106] L. Breiman, Random forests, *Math. Learn* 45 (2001) 5–32.
- [107] P. Probst, M.N. Wright, A.-L. Boulesteix, Hyperparameters and tuning strategies for random forest, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 9 (2019), e1301.
- [108] S.M. Maley, D.-H. Kwon, N. Rollins, J.C. Stanley, O.L. Sydora, S.M. Bischof, D. H. Ess, Quantum-mechanical transition-state model combined with machine learning provides catalyst design features for selective Cr olefin oligomerization, *Chem. Sci.* 11 (2020) 9665–9674.
- [109] G.A. Landrum, J.E. Penzotti, S. Putta, Machine-learning models for combinatorial catalyst discovery, *Meas. Sci. Technol.* 16 (2004) 270.
- [110] T. Janela, J. Bajorath, Simple nearest-neighbour analysis meets the accuracy of compound potency predictions using complex machine learning models, *Nat. Mach. Intell.* 4 (2022) 1246–1255.
- [111] S. Uddin, I. Haque, H. Lu, M.A. Moni, E. Gide, Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction, *Sci. Rep.* 12 (2022) 6256.



- [112] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, *Stat. Comput.* 14 (2004) 199–222.
- [113] A. Mammone, M. Turchi, N. Cristianini, Support vector machines, Wiley Interdiscip. Rev. Comp. Stat 1 (2009) 283–289.
- [114] H. Li, Y. Liang, Q. Xu, Support vector machines and its applications in chemistry, *Chemom. Intell. Lab. Syst.* 95 (2009) 188–198.
- [115] K. Heikamp, J. Bajorath, Support vector machines for drug discovery, *Expert Opin. Drug Discov.* 9 (2014) 93–104.
- [116] J. Wainer, P. Fonseca, How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms, *Artif. Intell. Rev.* 54 (2021) 4771–4797.
- [117] P. Birzhandi, K.T. Kim, H.Y. Youn, Reduction of training data for support vector machine: a survey, *Soft Comput.* 26 (2022) 3729–3742.
- [118] A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial, *Computer* 29 (1996) 31–44.
- [119] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, *Nature* 323 (1986) 533–536.
- [120] S. Curteanu, H. Cartwright, Neural networks applied in chemistry. I. Determination of the optimal topology of multilayer perceptron neural networks, *J. Chemom.* 25 (2011) 527–549.
- [121] M.H.S. Segler, T. Kogej, C. Tyrchan, M.P. Waller, Generating focused molecule libraries for drug discovery with recurrent neural networks, *ACS Cent. Sci.* 4 (2018) 120–131.
- [122] Z. Wu, J. Wang, H. Du, D. Jiang, Y. Kang, D. Li, P. Pan, Y. Deng, D. Cao, C.-Y. Hsieh, T. Hou, Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking, *Nat. Commun.* 14 (2023) 2585.
- [123] E. Burello, D. Farrusseng, G. Rothenberg, Combinatorial explosion in homogeneous catalysis: screening 60,000 cross-coupling reactions, *Adv. Synth. Catal.* 346 (2004) 1844–1853.
- [124] A.R. Khataee, M.B. Kasiri, Artificial neural networks modeling of contaminated water treatment processes by homogeneous and heterogeneous nanocatalysis, *J. Mol. Catal. A Chem.* 331 (2010) 86–100.
- [125] A.F. Zahrt, J.J. Henle, B.T. Rose, Y. Wang, W.T. Darrow, S.E. Denmark, Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning, *Science* 363 (2019) 5631.
- [126] S. Singh, M. Pareek, A. Changotra, S. Banerjee, B. Bhaskararao, P. Balamurugan, R.B. Sunoj, A unified machine-learning protocol for asymmetric catalysis as a proof of concept demonstration using asymmetric hydrogenation, *Proc. Nat. Acad. Sci. USA* 117 (2020) 1339–1345.
- [127] A.V. Brethomé, R.S. Paton, S.P. Fletcher, Retooling asymmetric conjugate additions for sterically demanding substrates with an iterative data-driven approach, *ACS Catal.* 9 (2019) 7179–7187.
- [128] T. Hayashi, K. Yamasaki, Rhodium-catalyzed asymmetric 1, 4-addition and its related asymmetric reactions, *Chem. Rev.* 103 (2003) 2829–2844.
- [129] B. Owen, K. Wheelhouse, G. Figueredo, E. Özcan, S. Woodward, Machine learnt patterns in rhodium-catalysed asymmetric Michael addition using chiral diene ligands, *Results Chem.* 4 (2022), 100379.
- [130] J.H. Friedman, Stochastic gradient boosting, *Comput. Stat. Data Anal.* 38 (2002) 367–378.