Exploring the potential of machine learning as a tool for chemical synthesis using SHAP

Peri Yerlikaya
Section de Chimie et Génie Chimique (SCGC), EPFL
Laboratory of Artificial Chemical Intelligence (LIAC), ISIC, EPFL
Laidlaw cohort 2023/2024
peri.yerlikaya@epfl.ch

Abstract

Recent years have seen the uprising of artificial intelligence within the scientific community both as a tool and a research topic. Studies on a wide range of subjects have shown that artificial models can outperform human experts, but how? Asking ourselves how a machine furnishes the decisions it makes is extremely pertinent, and to answer that question, we may use eXplainable Artificial Intelligence (XAI). The efficiency of the artificial models motivated the scientific community to start incorporating machines in various domains, such as chemistry, to see whether they could lead to improved results. However, while machine learning at the service of chemistry is actively being developed today, certain factors such as lack of interpretability and restricted availability of data for training slow its implementation. By applying select techniques from XAI to models trained with a data set taken from real-world experiments, this project aims to emphasize the suitability of machine learning to solve challenges in chemical synthesis. Additionally, the importance of methods that enable humans to understand, trust and manage artificially intelligent partners will be highlighted. All employed code can be found publicly at `https://github.com/yerlikayaperi/shap-for-chem`.

1 Introduction

The results of an experiment conducted in 2017 by Duros et al. [1], where a robot with an active machine learning algorithm was tasked to form and crystallize a specific polyoxometalate cluster, showed that the machine's predictions were on average more accurate and its efficiency higher than those of human chemists. This robot was a Bayesian reaction optimizer which proposed what it deemed to be ideal reaction space configurations by exploring areas of uncertainty and exploiting available knowledge within its training set. Although the mentioned framework resembles one a chemist would follow, the intuition of the machine oftentimes differs from the human specialist's. The gap between human and machine was brought to the fore in this experiment as the globally optimal conditions were unfamiliar to more experienced chemists, who thus tended to investigate in incorrect directions while the robot discovered the uncommon configurations quite early which benefited laboratory limitations [2]. Bayesian optimization was therefore recognized as an advantageous algorithm to implement in machine learning models designed for chemical usage, yet is still underutilized today.

Several more or less complex models were trialed in respect to their proficiency as chemical tools; notably ChemCrow [3] which is a Large Language Model capable of solving various types of chemical tasks the user inputs in natural language format (in other words, a Chat GPT-like robot specializing itself in chemistry). While rapid advancements occur within the field of artificial chemical intelligence, there have so far been very few implementations of the developed tools in industry and laboratories that claim these systems remain challenging to integrate in their workflow. Notice that this is not

exclusive to the field of chemistry—in general, artificial intelligence is regarded rather warily despite its efficiency as it tends to come at the detriment of its interpretability. From the new generation of increasingly complex machine learning models has emerged the need for eXplainable Artificial Intelligence (XAI) to maintain high performance without losing all understanding of the model's internal mechanism.

There are two sides to XAI: model understandability and human understandability. For a model to be understandable to a human, its function must be intelligible without the need to explain its internal structure or the algorithmic mean by which it processes data (corresponding to model understandability)[4]. Going against the famous Occam's razor, in the case of artificial intelligence the simplest and most easily explainable solution is often not the best one as will be shown in this report; thus the danger of creating and using predictions without legitimate justification arises. And for chemistry where resources are limited and compounds can be dangerous to manipulate, every decision must be thoroughly explained before it is undertaken in the laboratory.

Hence, XAI is vital to artificial chemical intelligence. In this project we will be applying a specific method called SHapley Additive exPlanations (SHAP)[5] proposed by Lundberg and Lee in 2017 to interpret the results that our machine learning models will provide and translate them to visual and natural language explanations aimed to be comprehensible by all. With these explanations, our goal is to reveal what the models have learned and how they apply it in their predictions to ultimately bridge the gap between data and modeler.

2 A brief overview of SHAP

SHAP is a model-agnostic interpretation method meaning that it is applicable to all machine learning models regardless of their complexity. It uses the concept of Shapley values introduced by Lloyd S. Shapley in 1953 which is a mathematical way of fairly dividing payout among players in a cooperative game. Shapley equates fairness to egalitarianism in his work by defining a marginal contribution that, for a certain player in a game, is the difference between the value of a coalition with the player and the value of the same coalition without the player[6]. As such, Shapley values are the weighed average of a player's marginal contributions to all possible coalitions within the observed game and can be formalized as below:

$$\phi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(N-|S|-1)!}{N!} \left(v(S \cup \{j\}) - v(S) \right) (1)^6$$

This equation is quite complex, but becomes less intimidating when broken down to its core elements. First of all, $v : P(N) \to R$ is a value function mapping all possible combination of *N* players to a numerical value representing the payout of the game, therefore $(v(S \cup \{j\}) - v(S))$ is the marginal contribution of player j to coalition S. The second term of the product determines the weight of a marginal contribution as the entire formula is a weighed sum over all possible coalitions not including j. Once the separate elements are identified, this equation is recognized to fit the definition given for Shapley values.

Following the basis that a prediction made by a computer can be viewed as a coalition game by considering each feature value of an instance (i.e. parameter) as a "player" in a game, with the "payout" becoming the value predicted by the machine, one can apply the same mathematical rules used for Shapley values to interpret the outputs of an artificial intelligence. The equation is thus adapted to explain a prediction:

$$\phi_j^{(i)} = \sum_{S \subseteq \{1 \dots p\} \setminus \{j\}} \frac{|S|!(p-|S|-1)!}{N!} \left(\int f(x_{S \cup j}^{(i)} \cup X_{C \setminus j}) \, dP_{Xc \setminus j} - \int f(x_S^{(i)} \cup X_c) \, dP_{Xc} \right) (2)^5$$

(2) has the same format as *(1)*, however the previous value function *v* is replaced by a more complex expression to fit the case of a specific model *f* and a particular feature value $x_j^{(i)}$ corresponding to a feature j. SHAP's approach is to treat the values X_C of features C not in the coalition *S* (i.e. the features forming C are absent in the input features *S* to predict a value $x^{(i)}$ as random variables and integrating over their distribution. Once this particularity has been set, the rest of the equation mirrors *(1)* in weighing marginal contributions and summing them for all possible combinations of p features not including j.

As SHAP values are simply Shapley values with a specific value function and game definition, they obey the same mathematical axioms: *Efficiency, Symmetry, Dummy* and *Additivity*. Each principle holds a certain implication in respect to the interpretation of SHAP values, so they will briefly be covered below.

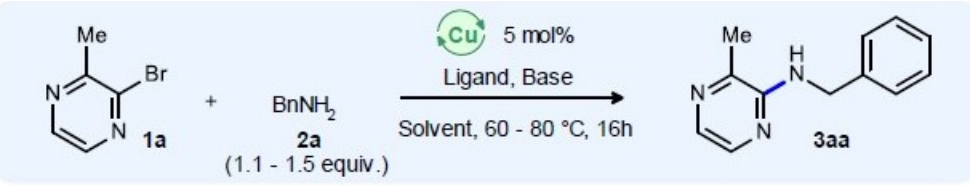
1. **Efficiency** states that SHAP values must total the difference between the prediction of a specific point and the average expected prediction of the set it is a part of. Therefore this axiom guarantees that attributions are on the scale of an output, rendering them easier to understand; it is in fact present in most XAI methods such as Local Interpretable Model-agnostic Explanations (LIME), to cite one.
2. **Symmetry** expresses that feature order is irrelevant, implying that if two features contribute equally, they will have the same SHAP value. This principle is not only essential to accurately interpret the order of SHAP values, it is also pleasantly intuitive.
3. **Dummy** finds its name from the 'dummy' variables it refers to: the features not influencing the prediction which will receive a SHAP value of 0. Unused features receiving a null attribution is another relatively obvious axiom.
4. **Additivity** entails that additive value predictions correspond to additive SHAP values. It is an especially powerful principle for machine learning models that include several separate systems in them, such as a Random Forest Regressor constructed with quite a number of dissociated trees and a final regression task, since it ensures that the SHAP values for each separate part of the model can be computed on its own and ultimately averaged to obtain values for the entire model.

Despite the existence of various XAI methods, this project finds that SHAP is particularly appropriate for the task it aims to accomplish. By decomposing the final prediction into contributions of each attribute, SHAP provides a consistency between its values and the concrete outputs of the machine which is not found in other techniques (LIME, for example, creates a surrogate model around the unit whose prediction one aims to understand, limiting itself to arbitrary local explanations). In chemical synthesis, approximating the use of a certain reagent to a positive or negative numerical contribution leading to a numerical solution (the yield) is pertinent to simplify the task the model is performing and thus gain knowledge on how it is capable of reaching correct conclusions occasionally faster than human chemists.

3 Overview of the data set

3.1 Presentation of the experiment

A certain amount of data is required in order to successfully train and test a machine learning model, which is why this project will be focusing on an experiment conducted in 2023 by Braconi and Godineau[7] where **Reaction 1** was investigated via Bayesian optimization.



Reaction 1[7]

To increase the yield of **3aa**, several parameters of this reaction can be adjusted: they are detailed in the **Appendix**. Unfortunately, the conversion of **1a** is a relatively time-consuming process taking around 16 hours, which greatly limits the number of experiments that can be conducted within a set period. The goal of this study is thus to discover the best reaction space in the least amount of tries possible, and to do so Braconi and Godineau decide to integrate a Bayesian optimizer (BO) into their workflow.

Based on the reaction space, the objectives to achieve, the number of experiments per round and the overall experimental budget defined by the chemists, the BO plots an objective function and returns

the first round of experiments to conduct. After this initialization the process becomes iterative as the chemists perform the proposed experiments and input their results, which leads to the BO adjusting its objective function and returning a new round of experiments until it either depletes the budget or reaches the set objective.

Ultimately, the model is used to produce eleven rounds of eight reactions each and the process repeated three times using different acquisition functions: Thompson Sampling (TS), Expected Improvement (EI) and Upper-Confidence-Bound (UCB). An acquisition function performs a select trade-off between exploration of the space and exploitation of the data within the BO in order to be most efficient, and the chosen three are among those that were found to be particularly adept at chemical tasks. The study provides us with 264 completed experiments and a maximum yield of **3aa** found to be 87.2% by the EI function.

3.2 Initial data exploration

Before any machine learning is involved, we would like to have an understanding of the reaction space to anticipate certain features' importance. Several python packages such as pandas, matplotlib.pyplot, seaborn and numpy were utilized to plot the graphics of this section, and for the sake of brevity some will be in the **Appendix**. As most parameters have several possible values, the format of box plot is found to be the most appropriate for clearer graphics and comprehension. As an example, here is the box plot representing the yield of the experiment as a function of the involved ligand:

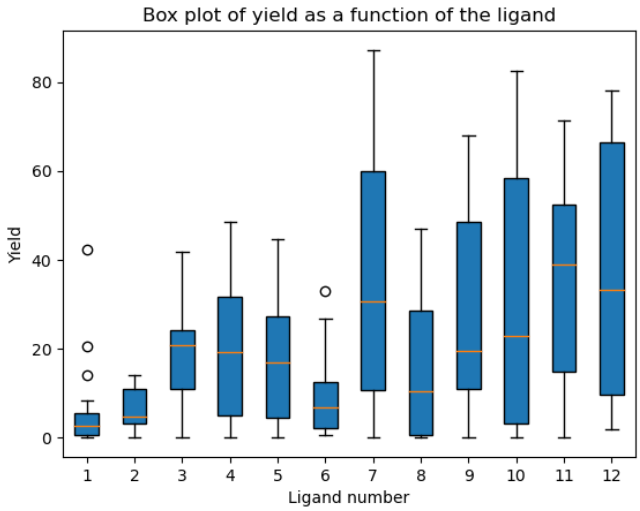


Figure 1: Box plot of the yield as a function of the ligand

From **Figure 1** it can be affirmed that the ligand leading to the experiment with the best yield is L7, that L1 and L2 performance are poorer than the other ligands' and that L11 and L12 seem to be the on average the most efficient. **Figures 5, 6** and **7** in the **Appendix** are of similar format but pertain to other features which highlight some other efficient parameters such as the bases K_3PO_4 and K_2CO_3 and the solvents DMSO and EtOH. Notice however that copper sources all seem relatively close in terms of performance.

Beyond features influencing the reaction, a comparison between the efficiency of the acquisition function is available in the initial study. It demonstrates that although the average yield percentage for each round as well as its increase through experiments is similar across all three functions, EI and UCB clearly perform better than TS[7]. While the maximum yield discovered via TS is 64.47%, UCB reaches 78.18% and EI 87.2%. It is additionally worth mentioning the combination leading to said highest yield:

Table 1: Experiment with highest yield	
Cu source	CuBr
Ligand	L7
Ligand equivalent	0.1
Base	K3PO4
Base equivalent	2.0
Solvent	DMSO
Molarity	1.0 M
Temperature	80 °C
BnNH2 equivalent	1.5
Round	9
Acquisition function	EI

These observations having been made with tools that we understand, we can trust their authenticity and compare them to those the artificial intelligence will make to convince ourselves of the soundness of its reasoning revealed with SHAP.

4 Training and observing the model

4.1 Choice of the method

In chemistry most of the latest studies involving artificial intelligence utilize Deep Learning (DL), which is a sub-section of machine learning methods based on artificial neural networks with multiple layers constructed in such ways to mimic the neural connections of the human brain. As their description entails, DL models are among the most complex, chemically efficient, but also resource consuming machine learning models to currently exist. However, to return to Occam's razor cited in the **Introduction**, while in machine learning the simplest solution will not always the best, there is no need to use an overly complex method if a simpler one produces satisfying results.

For this project, applying DL methods seems excessive given that the data set is constituted of 267 reactions only. A conscious management of resources is crucial as well, thus one would prefer utilizing a less 'expensive' model energy-wise while retaining a high accuracy. According to a study by Cao *et al.*[8], decision tree based ensemble methods abide by both criteria. Decision tree based ensemble methods are constructed using several decision trees which can be trained on select samples of the data set and averages their results to improve the predictive accuracy of the model they define. They represent on one of the most accurate learning algorithms available for chemical tasks as they can manage mixed or unbalanced data sets and missing values, therefore coping effectively with complex data. Additionally, they provide feature correlations which is pertinent for chemical reactions.

The model chosen for this project is as such sklearn's RandomForestRegressor, which is a meta estimator (i.e. combines the results of multiple predictions) fitting 100 decision trees on different randomized versions of the entire set and returning their average prediction. After One-Hot Encoding (OHE) the set, 80% of it is used to train the model, 10% to validate it and 10% to test it, and the optimal randomized split is determined using k-fold validation which leads to a model with a 78% predictive accuracy. In fact, this efficiency exceeded our expectations as the model reaches it within less than a minute: this is in part due to the size of the set, but is nevertheless appreciated if considered to be incorporated within an iterative workflow such as the one proposed by Braconi and Godineau[7].

Although the model is very satisfactory, some additional tests with different splits and models are conducted to see if it would have been possible to use a simpler method such as a Linear Regression or a General Additive Model (all available in the sklearn python library). Different ways of utilizing the set are also researched: whether OHE is necessary, the train set should be greater or not, etc. All graphical results are available in the **Appendix**, but the model with 78% remains the most efficient when looking at the entire set. Since features such as rounds and acquisition function did not directly influence the reaction itself they were not included in the model.

A separate model is also constructed using solely the experiments conducted with the EI acquisition function. By progressively training it using rounds in the same order as the initial study, it is possible

to observe whether this model agrees with the choice of experiments of the BO or not and even propose new combinations not present in the set.

4.2 Observation of the model without SHAP

It is possible to plot all 100 trees of the model, but studying them proves to be a tedious task not only due to their amount but also the number of features they include. For example **Figure 10** in the **Appendix** representing the first tree of the Random Forest contains over 210 nodes, rendering it hard to interpret in this form. One rather incomplete observation is that the features involved in the highest nodes tend to have the greatest impact on the prediction: some that are noted are the solvent DMSO, the bases K_2CO_3 and TBAO, and the 1.1 $BnNH_2$ equivalent. This implies that their SHAP values should stand out during discussion.

A single path in the tree does not include every feature since every possible combination is not present in the training set. However we can use this observation to stress the benefits of using a Random Forest instead of a singular tree: every tree will have different paths utilizing different features (this is not to say there will not be any similarities), therefore its output will differ depending on the features present or absent in the path corresponding to the input. By averaging all the outputs we make sure to take into account even the influence of the 'weaker' features neglected by certain trees.

Furthermore, the feature interactions discovered by the model can be discussed.

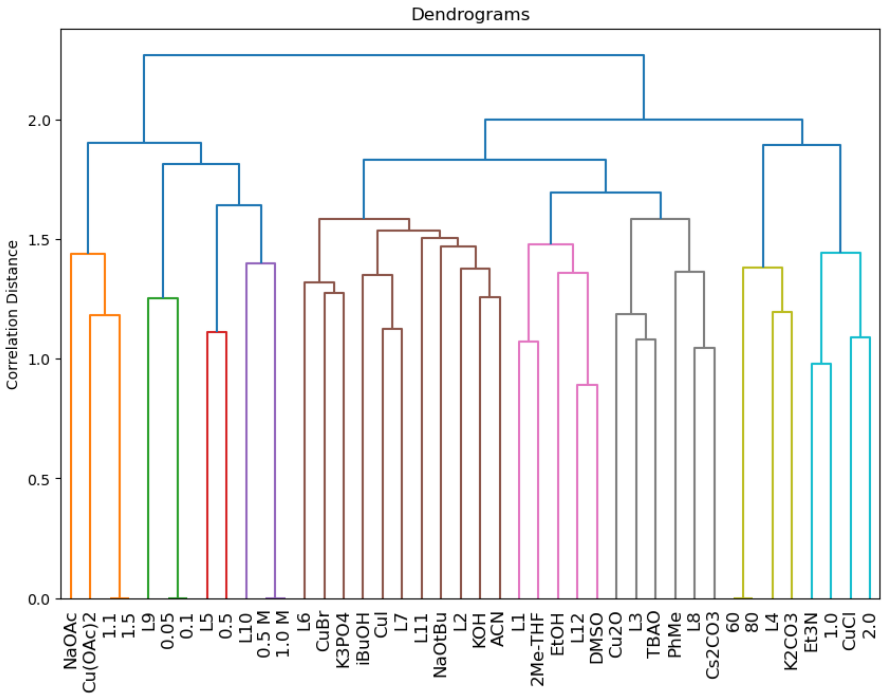


Figure 2: Dendogram representation of the model's correlation matrix

Due to OHE, the model easily attributes a 1:1 correlation between all pairs of features such as temperatures (in **Figure 2**, the yellow branch). It is also worth emphasizing that CuBr, K_3PO_4 and L7, which are respectively the copper source, base and ligand involved in the most satisfactory experiment, are clustered together (in **Figure 2**, the brown branches). Finally, the pink branch in **Figure 2** is quite interesting as it includes three very proficient features (DMSO, L12 and EtOH) grouped with two mediocre ones (L1 and 2Me-THF).

This is all the Random Forest package provides to explain its functioning. While these show the transparency of the model, they are relatively complex to discuss in their current form due to the size of the data set and the large amount of features involved. At this point, although the model has a high accuracy in predicting the yield of reactions in this specific reaction space, it would be beneficiary for

it to give a more detailed explanation of why it believes certain combinations would fare better than others.

5 Explaining the model's prediction with SHAP values

5.1 Visual explanations

The SHAP python package provides a number of tools to plot visual explanations of its values. One of them is called a waterfall plot which represents the local SHAP values for a specific prediction in the set.

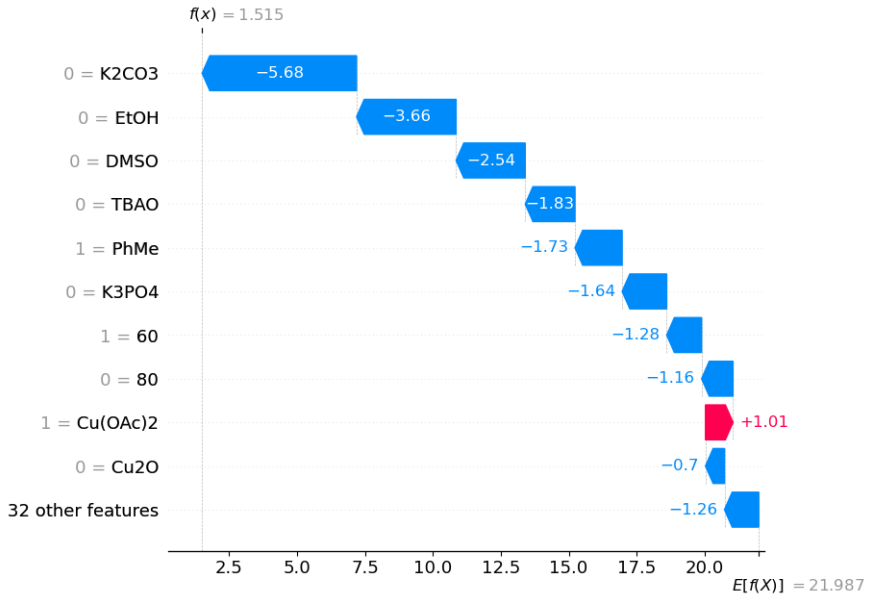


Figure 3: Waterfall plot of the prediction of the first point

Figure 3 shows the impact of the presence or the absence of features in the prediction of the yield *f(x)* as numerical deviations to the average predicted yield across the set *E[(f(X)]*. If a contribution is positive (resp. negative), it means the presence or the absence of the feature it represents increases (resp. lowers) the yield of the studied reaction. For example in **Figure 3**, the absence of K_2CO_3, EtOH, DMSO and TBAO has a strong negative impact on the yield of the reaction: in fact, these are features that were already recognized as having a high importance in the prediction. While this is only a local representation, a beeswarm plot can allow a global visualization of these impacts.

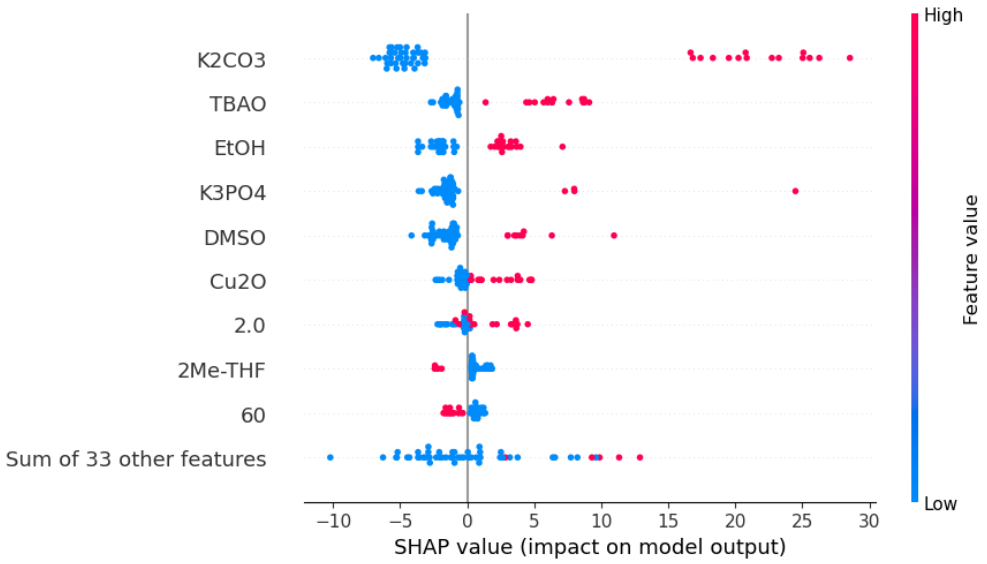


Figure 4: Beeswarm plot of the global feature importance

The advantage of this figure is that it includes whether the feature was present or absent at a certain experiment via colors (here a high feature value corresponds to its presence and a low feature value to its absence). **Figure 4** and **Figure 3** display a similar ranking of the most to least important features, which also correlates with the observations made on the model.

While these plots are visually easier to comprehend, they can be improved by introducing natural languages explanations that would directly provide a SHAP analysis instead of forcing the scientist to determine them themselves.

5.2 Natural language explanations

For this subsection, all the employed functions were written manually in python and tested on separate models. The aim was to produce short natural language explanations to the SHAP values of all the present features first on a local then a global scale.

```
shap_value_effect('K2CO3', 0)
```
Python
```
The SHAP value of K2CO3 for the experiment number 0 was -5.67856510845013.
The lack of K2CO3 in this experiment made the yield lower than average.
Predicted yield of this experiment = 1.5147999999999993
Actual yield of this experiment = 0.0
```
```
shap_value_effect('EtOH', 0)
```
Python
```
The SHAP value of EtOH for the experiment number 0 was -3.661597281426657.
The lack of EtOH in this experiment made the yield lower than average.
Predicted yield of this experiment = 1.5147999999999993
Actual yield of this experiment = 0.0
```

Illustration 1: Results of the shap_value_effect() function

Illustration 1 demonstrates the results of calling the *shap_value_effect()* function taking two arguments: the name of the feature to observe and the experiment to focus on. By simply inputting these arguments the method returns the corresponding SHAP value, its explanation, and the prediction as well as the actual yield of the concerned experiment. The explanation is concise and clear; significantly faster to comprehend than the previous plots.

```
shap_global('K2CO3')
```
Python

```
The mean SHAP value impact on the yield when K2CO3 is present is 21.802705896845396
The mean SHAP value impact on the yield when K2CO3 is absent is -4.798291795089791
Overall, the presence of K2CO3 makes the yield higher.
```

```
shap_global('PhMe')
```
Python

```
The mean SHAP value impact on the yield when PhMe is present is -2.7480043280102184
The mean SHAP value impact on the yield when PhMe is absent is 0.2998275149021799
Overall, the presence of PhMe makes the yield lower.
```

```
shap_global('2Me-THF')
```
Python

```
The mean SHAP value impact on the yield when 2Me-THF is present is -2.265597039930626
The mean SHAP value impact on the yield when 2Me-THF is absent is 0.8058906261358146
Overall, the presence of 2Me-THF makes the yield lower.
```

Illustration 2: Results of the shap_global() function

Illustration 2 represents the results of a second function, this time returning SHAP values on a global scale. Notice that both *shap_value_effect()* and *shap_global()* provide explanations that are specific to a single feature, which is a downside compared to the visual explanations, but it is possible to produce functions that compare the SHAP values of two features, rank them, etc. The opportunities are multiple; for the sake of this project, only a select few were investigated.

```
Maximum shap value per category :
Base : K2CO3
Solvent : DMSO
Cu source : Cu2O
Ligand : L11
Ligand equivalent : 0.1
Molarity : 0.5 M
Base equivalent : 2.0
BnNH2 equivalent : 1.1
Temperature : 60
--------------------------------------
Minimum shap value per category :
Base : KOH
Solvent : iBuOH
Cu source : CuI
Ligand : L8
Ligand equivalent : 0.05
Molarity : 1.0 M
Base equivalent : 1.0
BnNH2 equivalent : 1.5
Temperature : 80
```

Illustration 3: Maximum and minimum mean absolute SHAP values

Something to note in **Illustration 3** is that the model considers 60°C as a more influential factor than 80°C despite the two representing the only temperatures the reactions are tested at. This does not mean that experimenting at 60°C leads to higher yields than at 80°C—as seen in **Figure 4**—but that the model interprets a the detrimental effect of a lower temperature as more more important than the beneficial effect of a high temperature. Even in natural language explanations, SHAP values are influenced by both local and global effects which do not always align with human expectations. Nevertheless, there are no significantly incorrect explanations: the intuition of the robot aligns with an objective assessment of the data set.

6 Studying the Bayesian Optimizer's Expected Improvement function with SHAP

In total, the EI function returned 88 experiments it judged were most likely to lead to high yields. With these reactions it is possible to train a Random Forest Regressor following to observe whether it

can propose new combinations depending on SHAP values at each round and if these combinations were utilized or not by the BO. Before discussing the results, it is necessary to indicate that there were not enough data points for the model to have a high accuracy, however the SHAP values (especially in latter rounds) were very similar to those of the previous model.

Table 2: Best reaction at each round according to SHAP

Round	0	1	2	3	4	5	6	7	8	9	10
Solvent	EtOH	EtOH	EtOH	EtOH	EtOH	EtOH	EtOH	EtOH	EtOH	EtOH	DMSO
Cu source	Cu(OAc)2	Cu2O	Cu2O	CuI	CuCl	Cu2O	Cu2O	Cu2O	CuBr	CuBr	CuBr
Ligand	L5	L3	L5	L4	L4	L4	L9	L9	L7	L7	L7
Ligand equiv.	0.05	0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.1	0.1	0.1
Molarity	0.5 M	0.5 M	1.0 M	1.0 M	1.0 M	1.0 M	1.0 M	1.0 M	1.0 M	1.0 M	1.0 M
Base equiv.	2	2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	2
$BnNH_2$ equiv.	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
T [°C]	60	60	60	60	60	60	80	60	80	80	80

From **Table 2**, only the reaction found at Round 5 was actually performed during the study and had a yield of 45.36%, which is above the average yield obtained during Rounds 5 and 6[7]. In latter rounds, the proposed conditions are also recognized as beneficial to the yield according not only to the initial data analysis but also the results of the accurate model trained with the entire data set. There is thus a high probability that some of these experiments may lead to satisfactory yields if they were to be attempted.

A similar study could be done with the reactions obtained with the other acquisition functions, TS and UCB, but due to time constraints we focused on EI as its results were on average significantly superior to the rest.

7 Conclusion and outlook

By exploiting the data of Braconi and Godineau's article *Bayesian Optimization as a Sustainable Strategy for Early-Stage Process Development? A Case Study of Cu-Catalyzed C–N Coupling of Sterically Hindered Pyrazines*[7], several machine learning models were trained to predict the yield of a certain reaction depending on its conditions. The method which lead to the highest accuracy, 78%, was a Random Forest coupled with a regression task. To convince ourselves of the efficiency of the model at recognizing beneficial and detrimental features, we used visual and natural language SHAP to compare its analysis of the features to our exploration of the data set and found no discrepancy, legitimating the Random Forest's predictions. Furthermore, the initial study included a Bayesian Optimizer in its workflow to uncover chemical combinations leading to satisfactory yields; it was efficient, but did not explore the entire reaction space due to the belief that it found the highest yield possible: 87.2%. By iteratively training a Random Forest and determining the best features with SHAP, new reactions with potentially high rewards were discovered.

Overall, SHAP were shown to be very appropriate to not only explain complex machine learning models but also especially in chemistry, rank features by their importance and predict efficient chemical reactions. This XAI technique is rarely utilized in chemical task despite its simple implementation, which could be due to its publicly available representations being solely graphical and thus not always appropriate. In this project, functions were manually written to interpret SHAP values in a natural language format but one could consider using Large Language Models such as Langchain to automatize such tasks even further. In conclusion, artificial intelligence is a large domain which has the potential to significantly accelerate research in chemical synthesis and should continue to be experimented with.

8 Acknowledgments

I am very grateful to Prof. Philippe Schwaller and Dr. Geemi Wellawatte from the LIAC for their continuous support, kind help and valuable advice throughout this project. Thank you as well to the Laidlaw Foundation and the EPFL for giving me this incredible opportunity.

References

[1] Vasilios Duros, Dr. Jonathan Grizou, Dr. Weimin Xuan, Zied Hosni, Dr. De-Liang Long, Dr. Haralampos N. Miras, and Prof. Leroy Cronin. Human versus robots in the discovery and crystallization of gigantic polyoxometalates. *Angewandte Chemie - International Edition*, 56(36):10815–10820, 2017.

[2] Benjamin J. Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I. Martinez Alvarado, Jacob M. Janey, Ryan P. Adams, and Abigail G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590:89–96, 2021.

[3] Andres M. Bran, Sam Cox, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *arxiv*, 2023.

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[5] Scott Lundberg and Su in Lee. A unified approach to interpreting model predictions. *arxiv*, 2017.

[6] Lloyd S. Shapley. A value for n-person games. *Theory of Games II*, pages 307–317, 1953.

[7] Elena Braconi and Edouard Godineau. Bayesian optimization as a sustainable strategy for early-stage process development? a case study of cu-catalyzed c–n coupling of sterically hindered pyrazines. *ACS Sustainable Chemistry & Engineering*, 11(28):10545–10554, 2023.

[8] Dong-Sheng Cao, Jian-Hua Huang, Yi-Zeng Liang, Qing-Song Xu, and Liang-Xiao Zhang. Tree-based ensemble methods and their applications in analytical chemistry. *TrAC Trends in Analytical Chemistry*, 40:158–167, 2012.

A Appendix

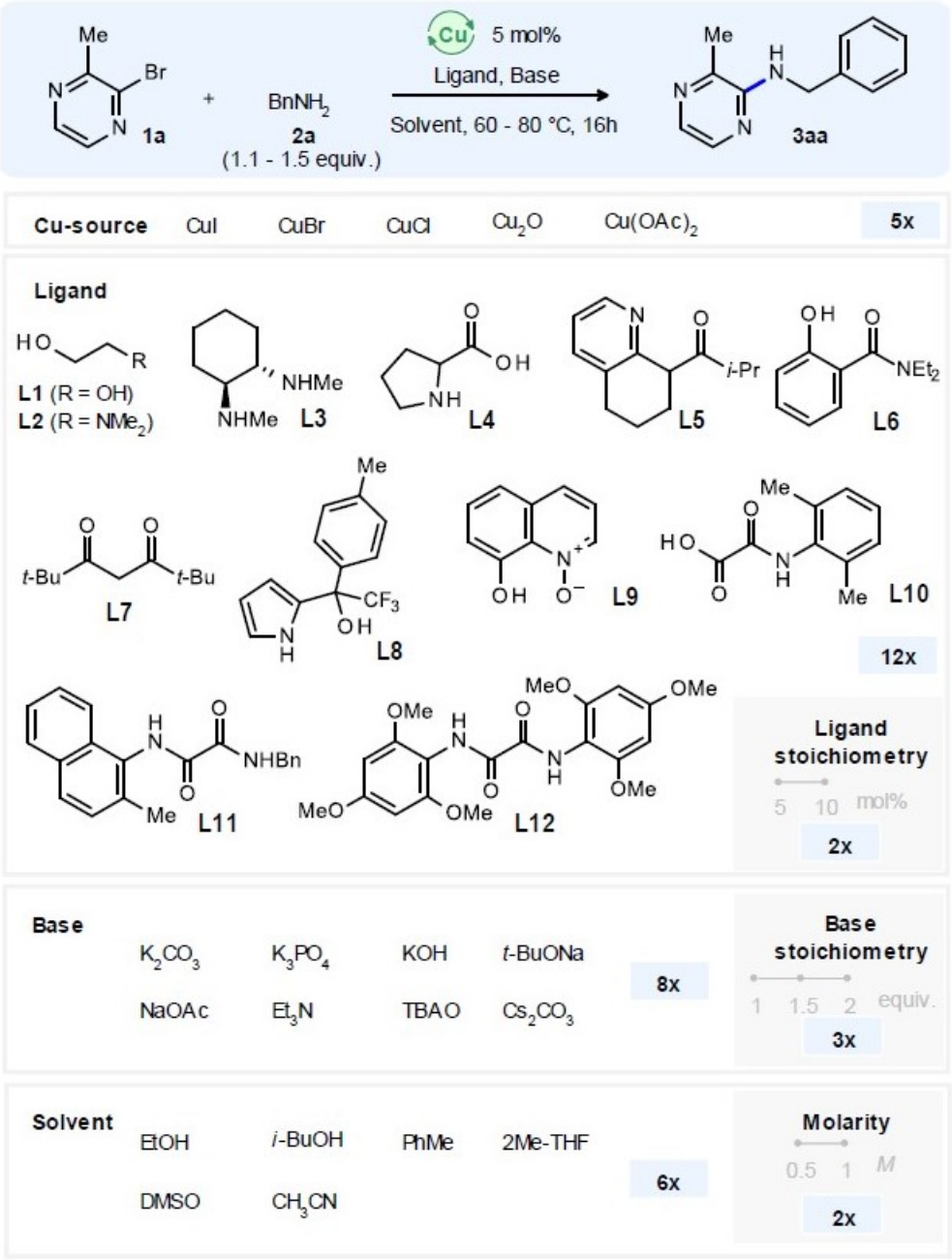


Illustration 4: reaction space[7]

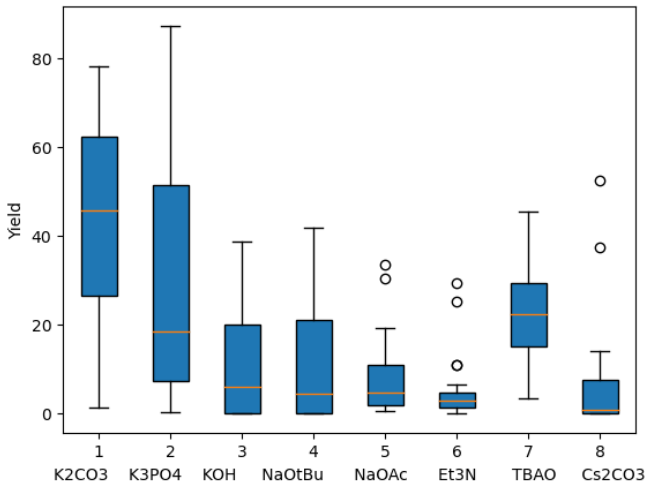


Figure 5: Box plot of the yield as a function of the base

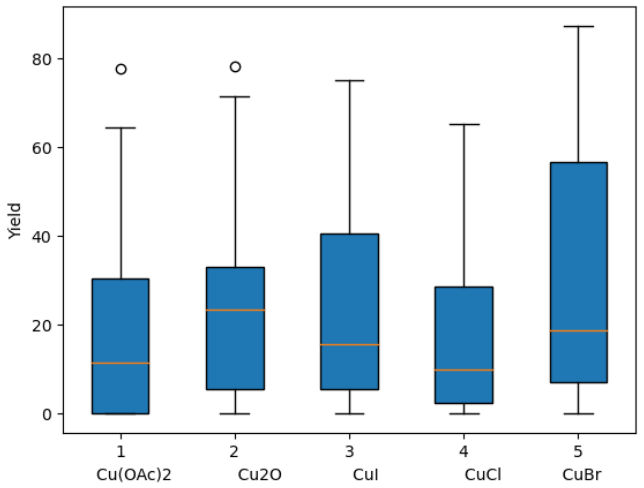


Figure 6: Box plot of the yield as a function of the copper source

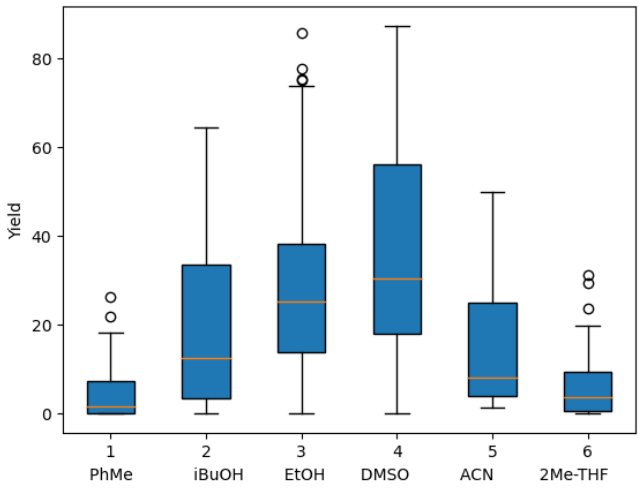


Figure 7: Box plot of the yield as a function of the solvent

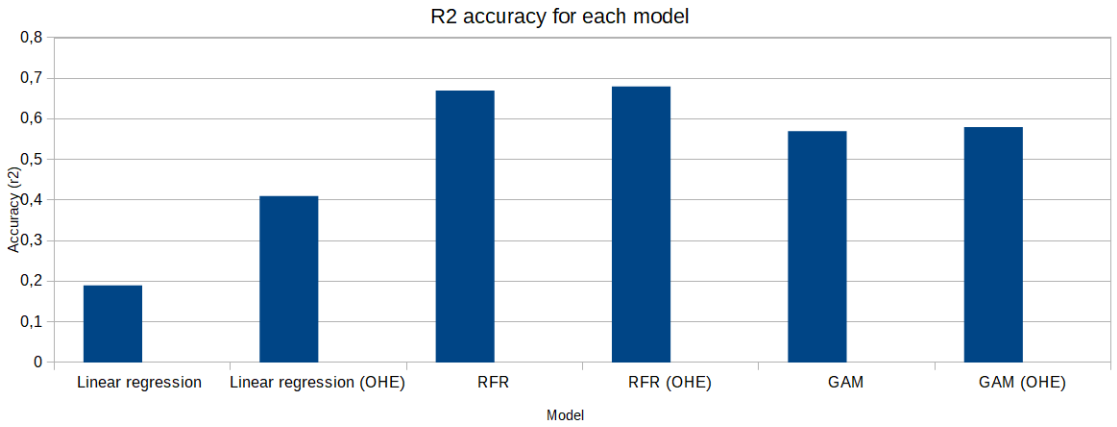


Figure 8: Accuracy of each tested model

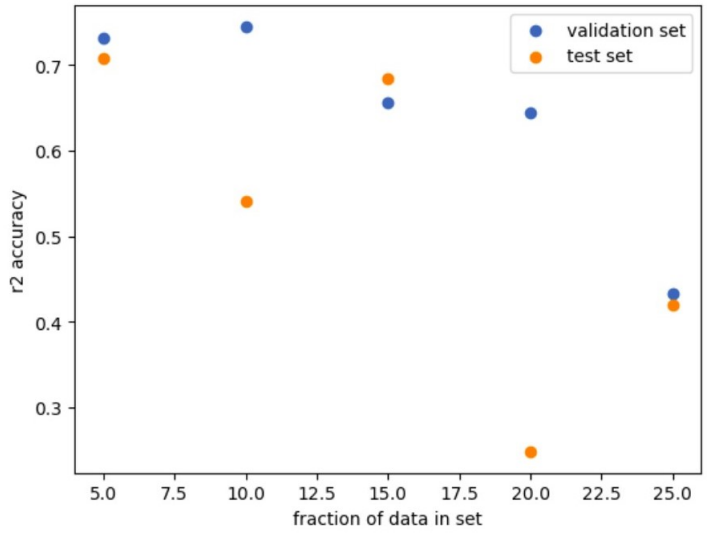


Figure 9: r2 accuracy of the Random Forest Regressor as a function of the % of data in the validation or test set

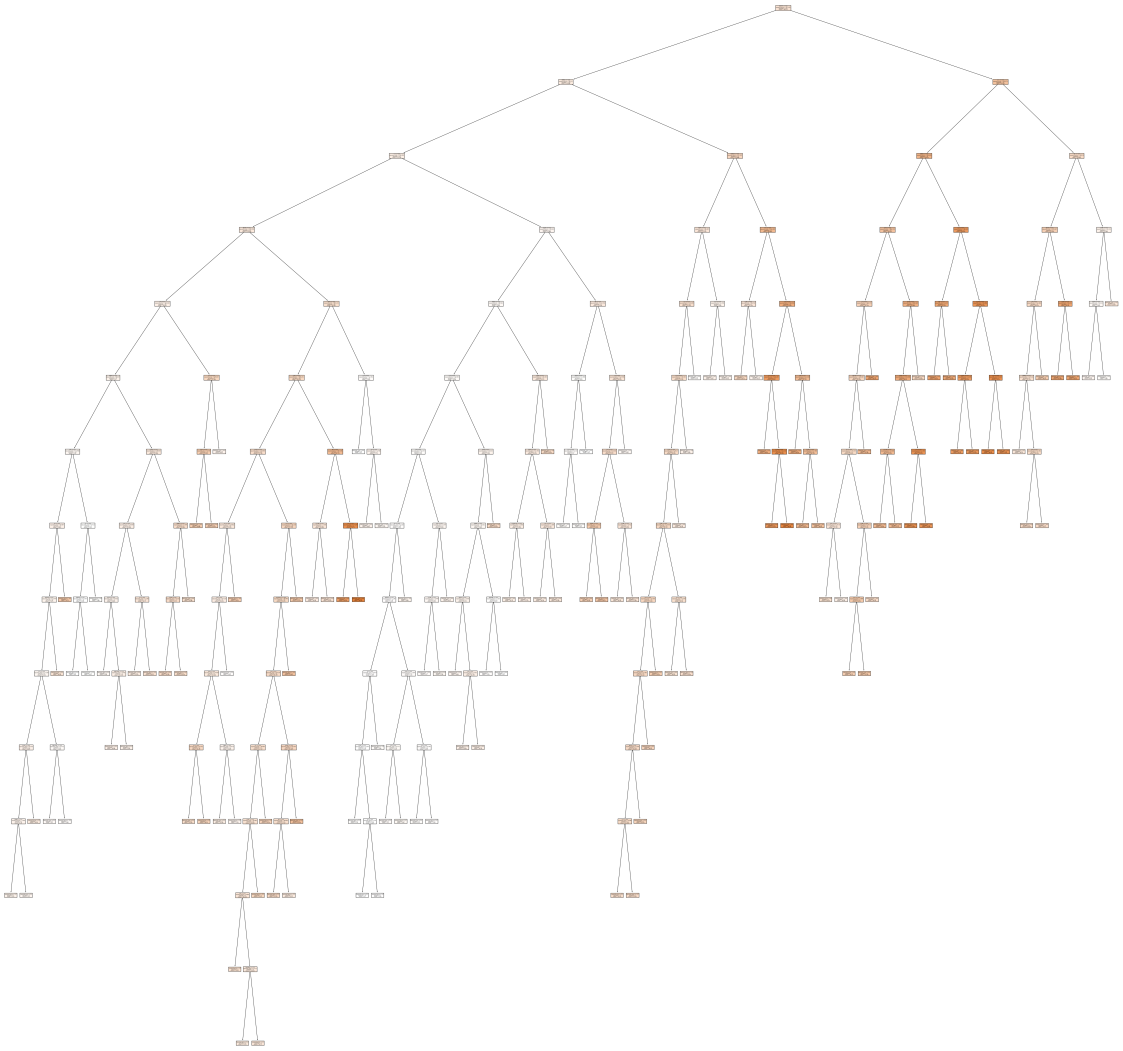


Figure 10: First decision tree of the Random Forest

Table 3: Features with the highest impacts on the prediction per round according to SHAP.

Rounds (cumulative)	Highest positive impact (present)	Highest negative impact (absent)	Highest positive impact (absent)	Highest negative impact (present)	Overall most influencial feature
0	$Cu(OAc)_2$	EtOH	60	$Cu(OAc)_2$	$Cu(OAc)_2$
1	80	TBAO	60	1.1	1.5
2	TBAO	TBAO	1.1	L3	TBAO
3	$Cu(OAc)_2$	TBAO	80	$Cu(OAc)_2$	$Cu(OAc)_2$
4	$Cu(OAc)_2$	0.5	80	$Cu(OAc)_2$	$Cu(OAc)_2$
5	$Cu(OAc)_2$	TBAO	CuCl	$Cu(OAc)_2$	$Cu(OAc)_2$
6	K_2CO_3	TBAO	1.1	KOH	KOH
7	K_2CO_3	TBAO	L5	2Me-THF	K_2CO_3
8	$Cu(OAc)_2$	K_2CO_3	0.05	$Cu(OAc)_2$	$Cu(OAc)_2$
9	L7	K_2CO_3	0.05	PhMe	L7
10	K_2CO_3	K_2CO_3	PhMe	PhMe	K_2CO_3