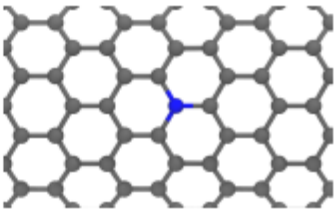
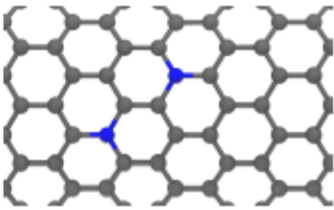
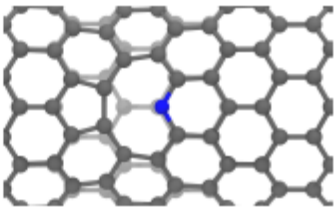


## Descriptors

Descriptors are very important in materials science/chemistry. We should know the geometry and other properties of the material or molecules but how we will tell that to a machine. The descriptors can be almost anything.

We did recently a study of HER (hydrogen evolution reaction) on N doped carbon nanotubes taking into account several defects. Overall, there was 8 different defects and several hydrogen configurations. Totally we did ca. 7000 DFT calculations. The output was the hydrogen binding energy. (Kronberg, Lappalainen, Laasonen, JPCC, 125, 15918 (2021)). In this project we used the Random Forest method and a very new Shapley analysis of the data.

**Table 1: Specification and illustration of the studied model NCNT systems. The abbreviations  $V_1$ ,  $V_2$  and SW denote a single vacancy, double vacancy and Stone–Wales rotation, respectively. For the Stone–Wales defect two distinct nitrogen positions resembling indole ( $N_{1a}$ ) and indolizine ( $N_{1b}$ ) structures were considered.**

N moiety	Defect	$(n, m)$	Image
Graphitic, $N_1$	None	$(14, 0)$ $(8, 8)$	
Graphitic, $N_2$	None	$(14, 0)$ $(8, 8)$	
Pyridinic, $N_1$	$V_1$	$(14, 0)$ $(8, 8)$	

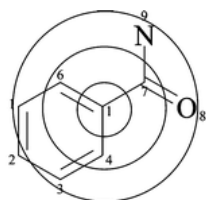
This project had rather complex descriptors. This example is not the easiest one, but it illustrates that the very different descriptors can be used.

Table 2: Mathematical formulation and explanation of all 25 input features used in the RF models. Note that all features except those inferring to adsorption-induced changes are calculated on the reference configurations, i.e. the NCNT structures before adsorption (NCNT + (n - 1)H). A glossary of auxiliary variables used in defining each feature is presented in Table S1 in the Supporting Information.

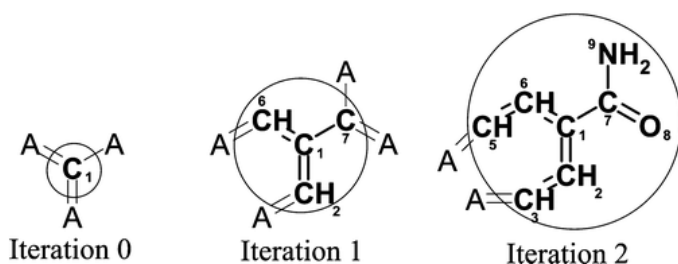
Feature	Definition	Unit	Description
$x_k$	$N_k/N_{\text{NCNT}}$	at%	Atomic concentration of $k = \text{N, V}$ or $\text{H}^a$
$\min d_k$	$\min_k \sqrt{r^2 \theta_k^2 + z_k^2}$	Å	Shortest curvilinear distance along NCNT surface between S and $k = \text{N}$ or $\text{H}^b$
$\min l$	$\min_k  \mathbf{R}_S - \mathbf{R}_k $	Å	Shortest S to $k \in \text{NN}$ bond length <sup>c</sup>
$\max l$	$\max_k  \mathbf{R}_S - \mathbf{R}_k $	Å	Longest S to $k \in \text{NN}$ bond length
RMSD	$\sqrt{\langle (\Delta \mathbf{R}_k)^2 \rangle}$	Å	Adsorption-induced root-mean-squared displacement of atomic positions
RmaxSD	$\sqrt{\max_k (\Delta \mathbf{R}_k)^2}$	Å	Adsorption-induced root-maximum-squared displacement of atomic position
$\chi$	$\arctan\left(\frac{\sqrt{3}m}{2n+m}\right)$	rad	Chiral angle of (n, m) NCNT
$\min \varphi_k$	$\min_{j \neq i} \arccos(\hat{\mathbf{u}}_{ik} \cdot \hat{\mathbf{u}}_{jk})$	rad	Smallest angle where $k = \text{S}$ or nearest N defines the vertex and $i, j \in \text{NN}$ of $k$
$\max \varphi_k$	$\max_{j \neq i} \arccos(\hat{\mathbf{u}}_{ik} \cdot \hat{\mathbf{u}}_{jk})$	rad	Largest angle where $k = \text{S}$ or nearest N defines the vertex and $i, j \in \text{NN}$ of $k$
$\alpha_k$	$\arccos(\hat{\mathbf{z}} \cdot \hat{\mathbf{v}}_k)$	rad	Angular displacement of S with respect to the nearest $k = \text{N}$ or H
$\overline{\text{CN}}_k$	$\sum_i \frac{\text{CN}_i}{\text{CN}_{i,\max}}$	-	Generalized coordination number of $k = \text{S}$ or nearest N with $i \in \text{NN}$
$\Delta \overline{\text{CN}}_k$		-	Adsorption-induced change in $\overline{\text{CN}}_k$
$Z$		-	Atomic number of the adsorption site
$M$	$2S + 1$	-	Spin multiplicity of the system
$q$	$n_{\text{val}} - (n_{\uparrow} + n_{\downarrow})$	e	Residual charge on the adsorption site (iterative Hirshfeld partitioning)
$\mu$	$n_{\uparrow} - n_{\downarrow}$	e	Spin polarization on the adsorption site (iterative Hirshfeld partitioning)
$E_g$	$E_{\text{LUMO}} - E_{\text{HOMO}}$	eV	Energy gap, for open-shell systems the SOMO-LUMO gap

<sup>a</sup>N, V, H = Nitrogen, vacancy, hydrogen; <sup>b</sup>S = Adsorption site; <sup>c</sup>NN = Nearest neighbor sites;

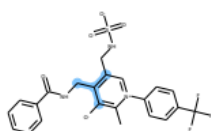
One interesting descriptor is Extended-connectivity fingerprint (ECFP). It is a systematic tool that list atoms environment in molecules. (Ref: Rogers and Hahn, *J. Chem. Inf. Model.* 2010, 50, 5, 742-754). The 0 level is the atom itself, the level 1 contains the atoms neighbors and so on.



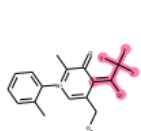
Considering atom 1 in benzoic acid amide



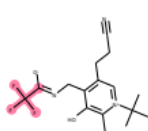
Next one can list all the different ECFP's of all the studied molecules. There are quite a few of them but surprisingly few. We did a project in which there were 7000 different molecules and we found 1024 ECFP4's



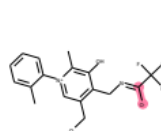
1010



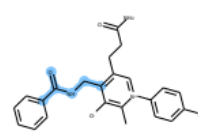
273



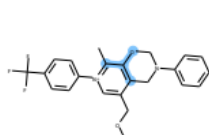
490



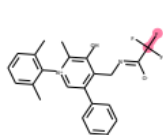
202



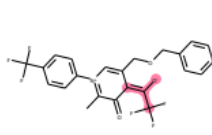
813



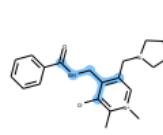
270



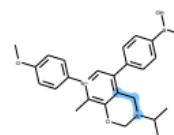
429



792



845



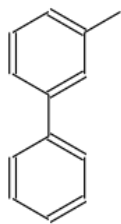
922

## Smiles

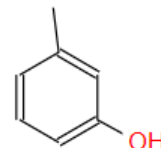
Is a very useful representation of molecules. It contains only letters and most chemical codes understand and can make SMILES. They usually can make also 3D coordinates.

Eg. benzene c1ccccc1

Cc2cccc(c1ccccc1)c2 =



Cc1cccc(O)c1 =

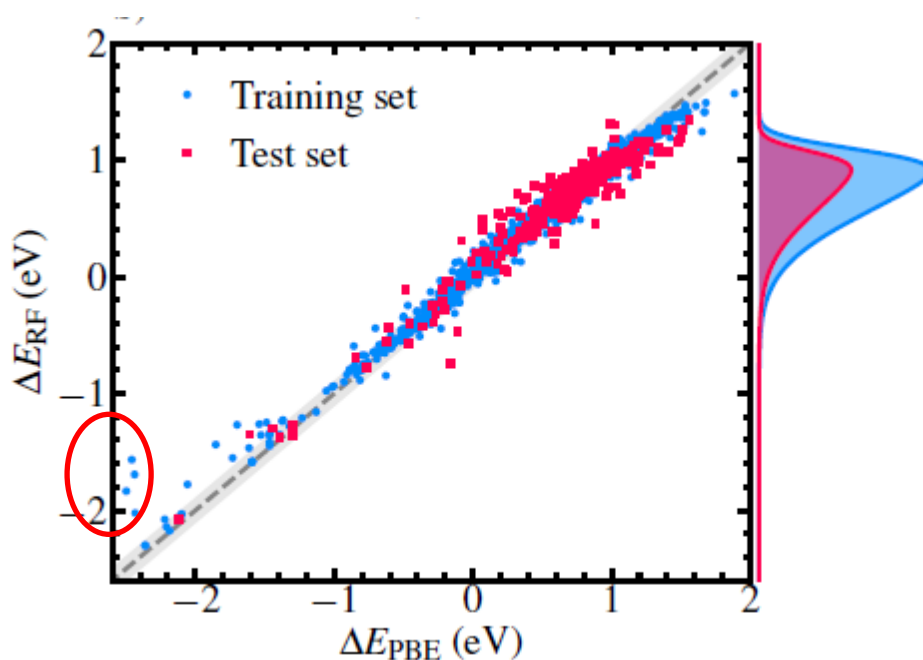


[https://www.cheminfo.org/flavor/malaria/Utilities/SMILES\\_generator\\_checker/index.html](https://www.cheminfo.org/flavor/malaria/Utilities/SMILES_generator_checker/index.html)

Smiles are not very easy for humans to read but they can be drawn easily.

## Results

The RF model learned the hydrogen binding (HER) data well. The parity plot compares the computed (DFT) values to the ML predictions.



As one can see, where there is a lot of data the learning is good and at the very negative values the scattering is larger. The accuracy of the trained data is below kcal/mol, which is better than the DFT accuracy. One can also see the effect of the size of the sample. We did some PBE0 calculations. Here the data set is much smaller and the learning errors are larger.

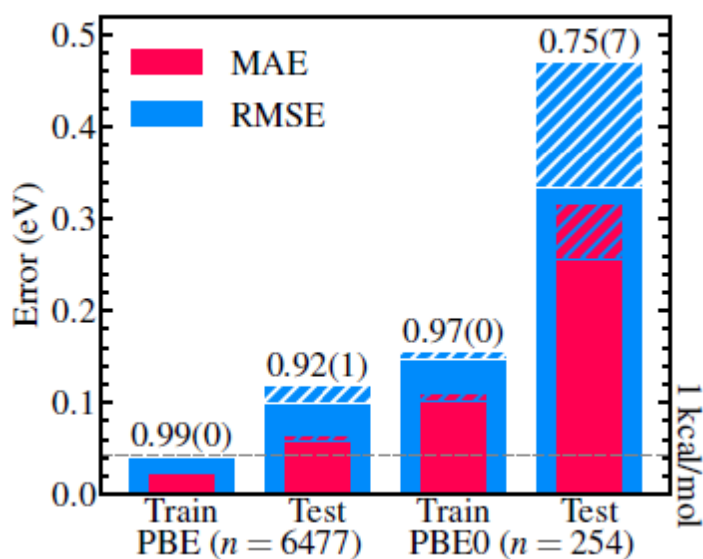
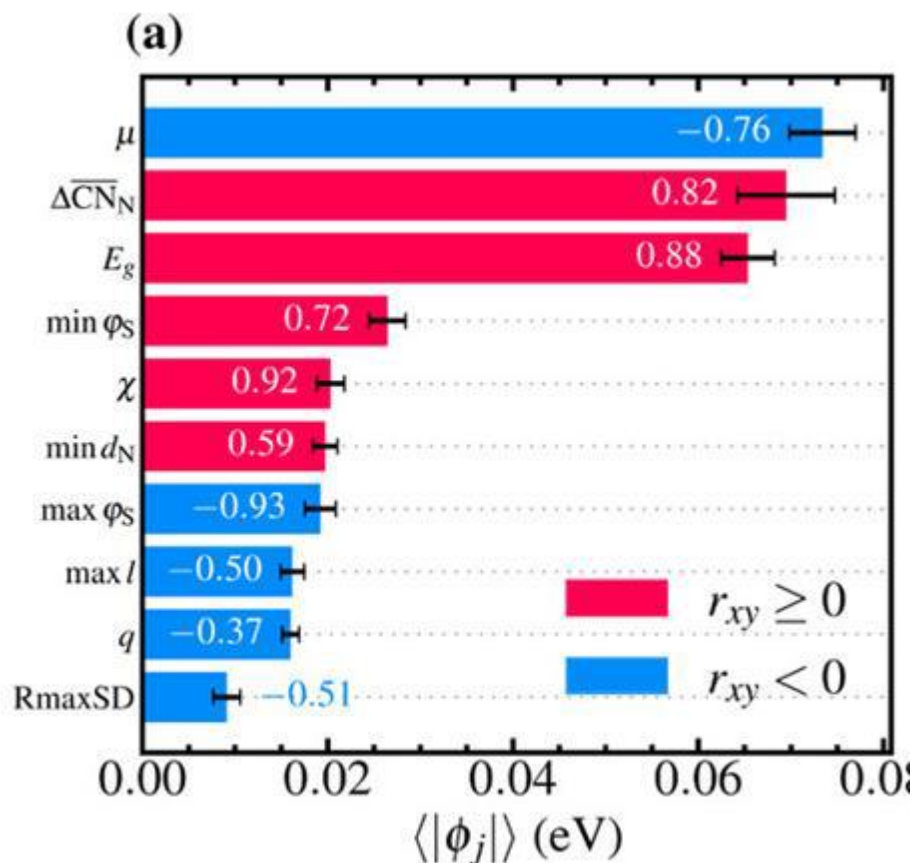


Figure 3: Unbiased generalization performance of the RF models based on 10-fold nested CV on the GGA and hybrid HF/DFT datasets. The solid bars denote a lower bound of the respective averaged errors while the hatched parts indicate the variability as twice the standard deviation across the outer CV folds. The average coefficients of determination with standard deviations are annotated above the bars. The limit of chemical accuracy is marked for reference by the dashed line.

The next deep question is how the descriptors contribute to the output. This is usually addressed on a rather superficial way. Typically, the methods like RF will return the weight of the descriptors. This is useful if some of the descriptors have low weights. Then one can reduce the descriptors and still get quite good predictions with less descriptors.



## Explainable AI, the Shapley analysis

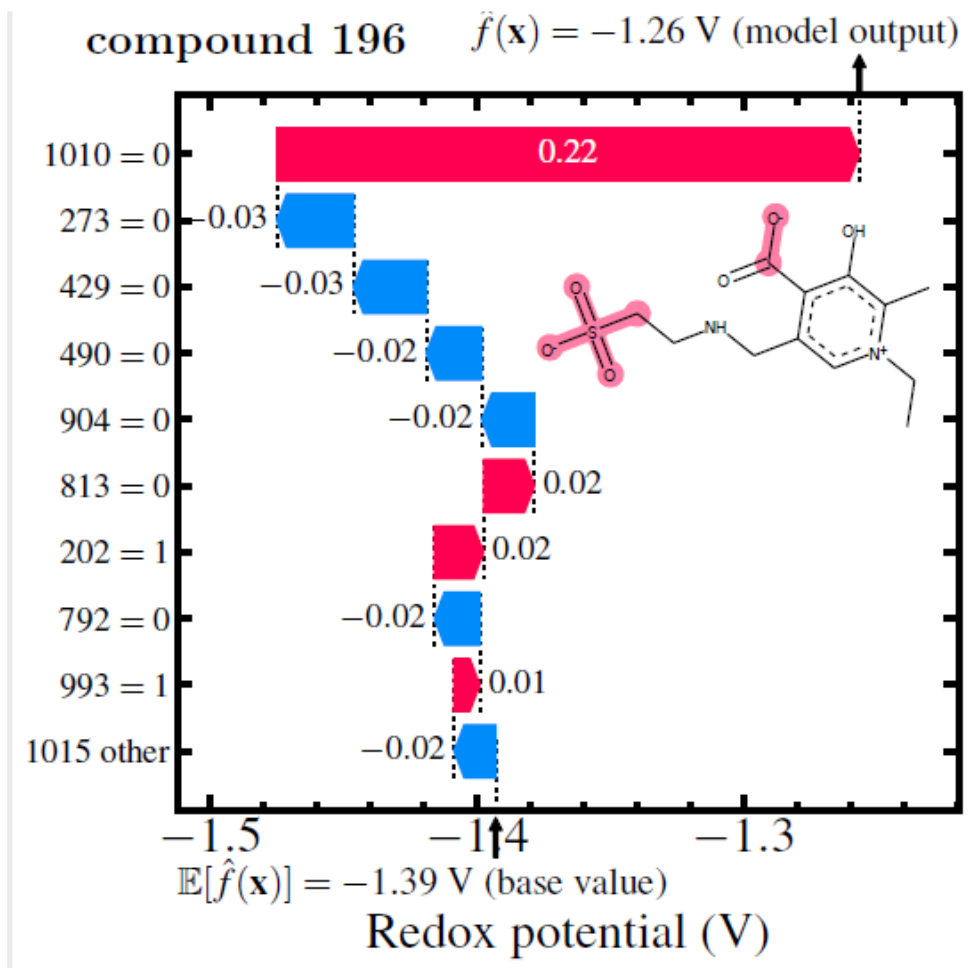
Rather recently a very interesting Shapley additive explanation (SHAP) methods has been introduced. It will approximate the model output with additive functions  $\phi$ , Shapley values. The ML predicted value  $f$  can be written as

$$f(x) = \langle f \rangle + \sum_j \phi_j(f, x)$$

where  $\langle \rangle$  is the average of  $f$  and  $x$  are the descriptors. Even this looks very simple the computation of the Shapley values is complicated. The breakthrough publication is from 2017 (Lundberg, S. M.; Lee, S.-I. A unified approach to interpreting model predictions. Adv. Neural Inf. Process Syst. 2017; pp 4765–4774.)

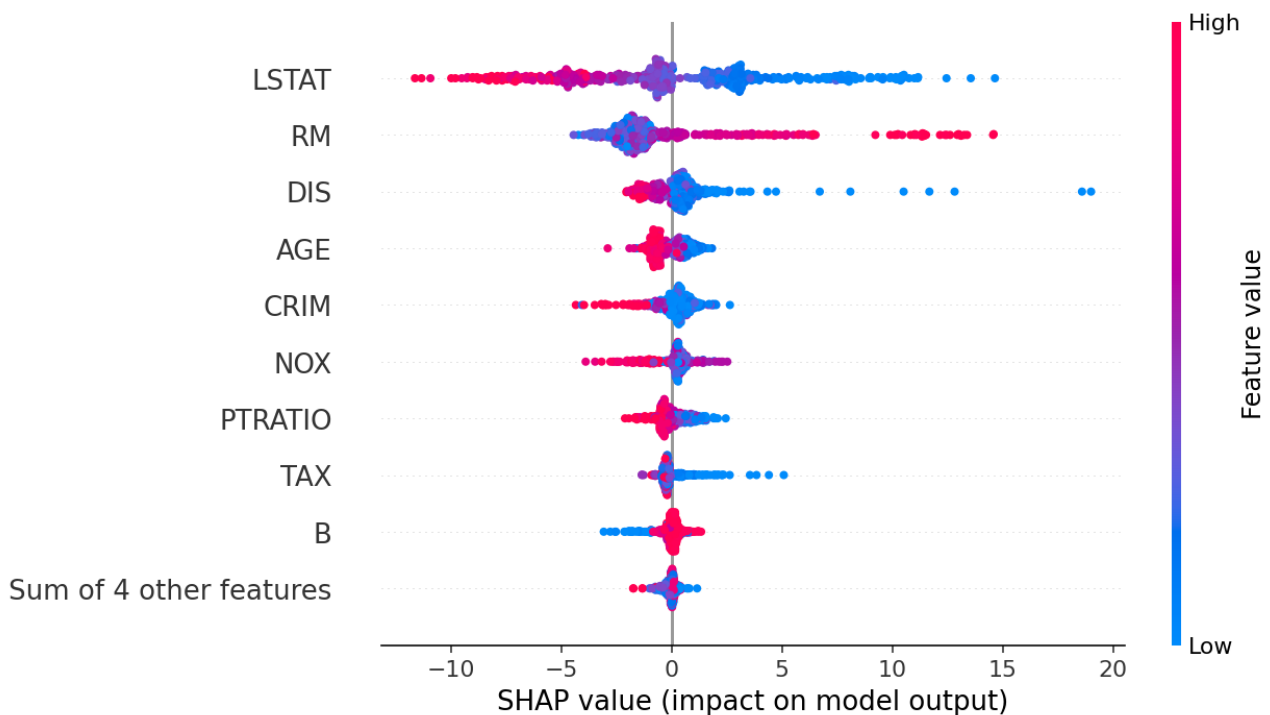
The SHAP analysis gives much more information of the ML procedure. We can analyse the individual descriptor contribution to the output. If we have chemically meaningful descriptors, we can learn a lot more

form the results. Below is an example of the molecules redox potential prediction. The numbers refer to the ECFP4 features in the molecules (0 means that they are not present and 1 that they are). Note that the 1015 lowest weight descriptors have very small contribution and the descriptor 1010 has very large contribution.



The SHAP analysis has results to a new subfield of ML, the explainable artificial intelligence (XAI). There are several problems where it is very useful to understand where the ML predictions come from. Clearly materials development projects belong to this class.

In SHAP we can also analyse features role in general. Below if the feature LSTAT has high value (red) it will have a negative contribution (and vice versa). The feature RM has an opposite effect and feature B has little effect.



## Unsupervised Learning

We have now several projects related to molecular clustering. The main idea is to rationalize chemical reactivity.

### Clustering

But first focus on clustering. At low dimensions we are good at seeing clusters. One of the simplest clustering algorithm is KMeans. It will find the centers of the clusters.

```

from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics.pairwise import pairwise_distances_argmin

np.random.seed(0)

batch_size = 45
centers = [[2, 2], [-2, -2], [2, -2]]
n_clusters = len(centers)
X, labels_true = make_blobs(n_samples=3000, centers=centers, cluster_std=0.9)

k_means = KMeans(init="k-means++", n_clusters=3, n_init=10)
k_means.fit(X)

k_means_cluster_centers = k_means.cluster_centers_
k_means_labels = pairwise_distances_argmin(X, k_means_cluster_centers)

fig = plt.figure(figsize=(4, 4))
fig.subplots_adjust(left=0.02, right=0.98, bottom=0.05, top=0.9)
colors = ["#4EACCE", "#FF9C34", "#4E9A06"]

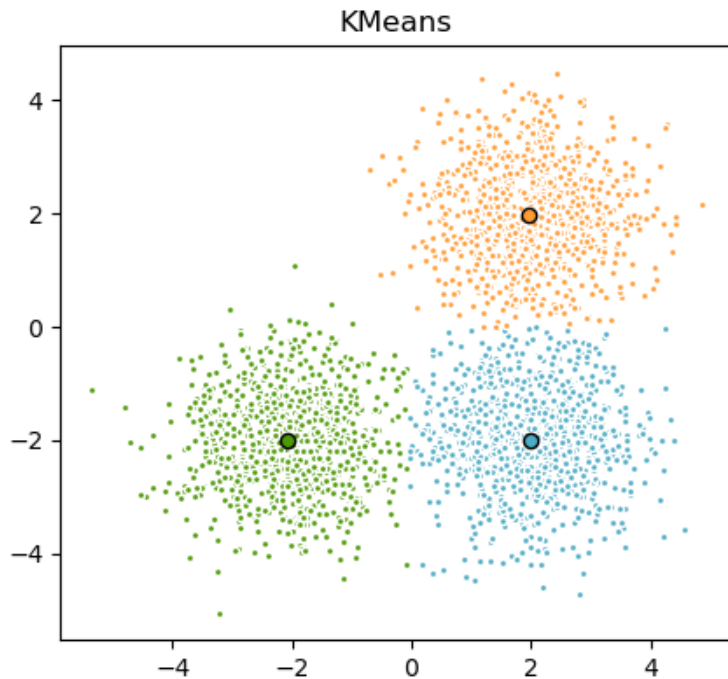
```



```

# KMeans
ax = fig.add_subplot(1, 1, 1)
for k, col in zip(range(n_clusters), colors):
    my_members = k_means_labels == k
    cluster_center = k_means_cluster_centers[k]
    ax.plot(X[my_members, 0], X[my_members, 1], "w", markerfacecolor=col,
            marker=".")
    ax.plot(cluster_center[0], cluster_center[1], "o",
            markerfacecolor=col, markeredgecolor="k", markersize=6,)
ax.set_title("KMeans")

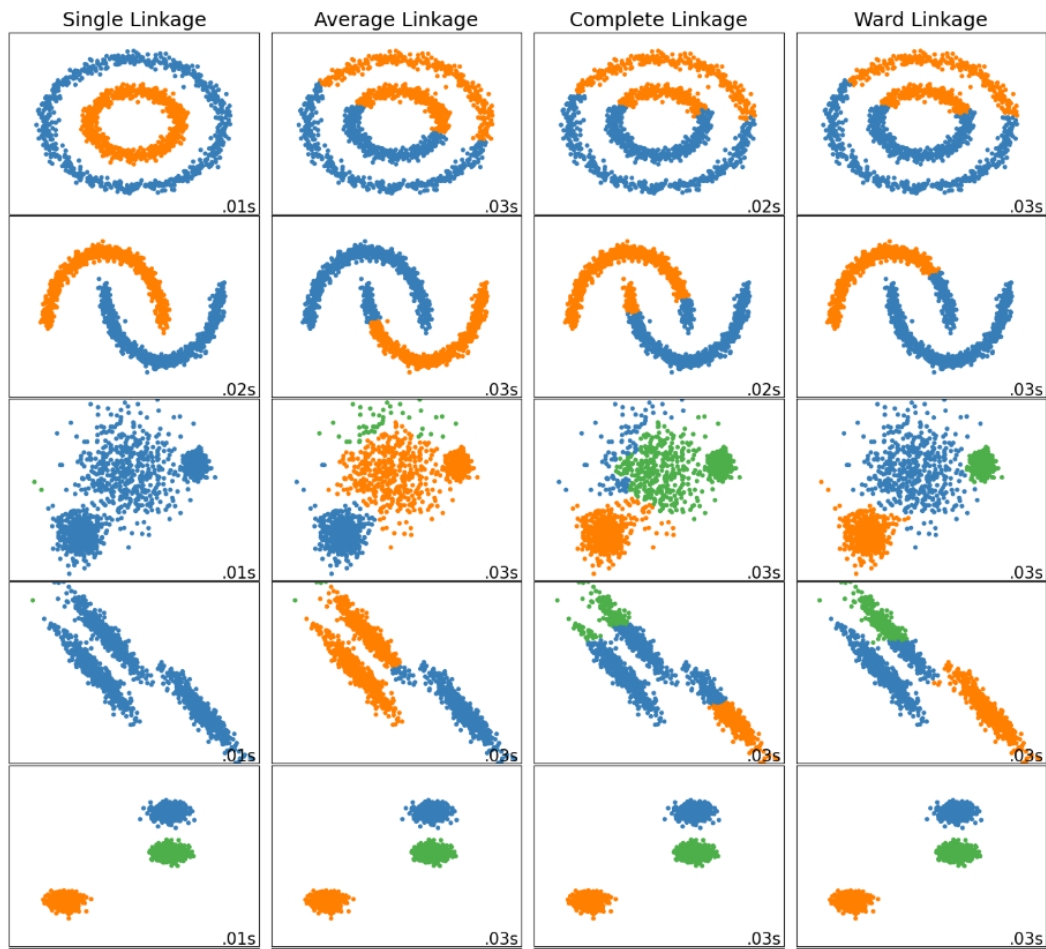
```



This is too simple. The clusters can and will have more complex shape. One more sophisticated clustering method is [AgglomerativeClustering](#) with different linkage methods. (Details in sklearn manual)

[AgglomerativeClustering\(linkage="ward", "complete", "average", "single"\)](#)

There is probably no single method to find nice clusters in all cases. Almost any method will find isolated clusters.



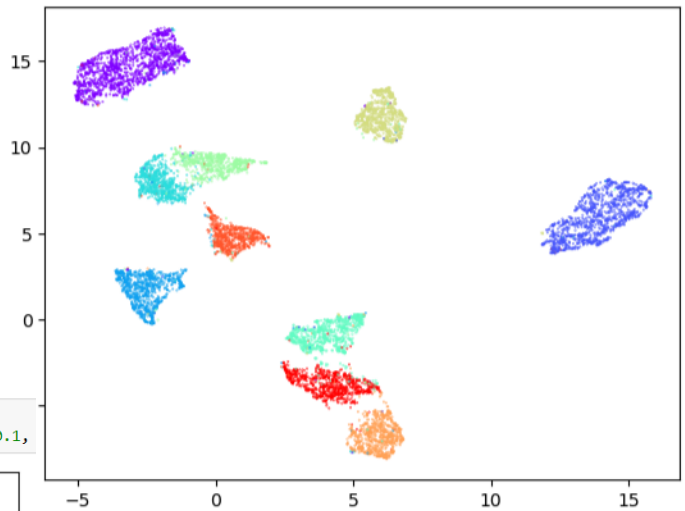
In real problems we have several descriptors, easily 256 to 1000. We can of course find clusters in 1000 dimensional space but it is very difficult to learn anything from this. We need dimensional reduction.

Example: handwritten numbers, left example of the data. (data set 9300 numbers, 16x16 pixels), right UMAP clustering. The colors are the correct numbers. This is a good clustering. Below is the Kmeans clustering of the UMAP data. Note that the colors are mixed, not good clustering.

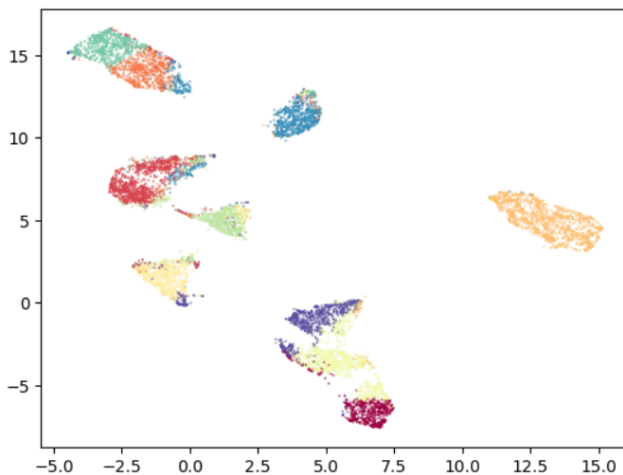
Uncorrupted test images



```
s2_embedding = umap.UMAP(random_state=14,n_neighbors=10).fit_transform(X)
plt.scatter(s2_embedding[:, 0], s2_embedding[:, 1], c=y.astype(int), s=0.1,
```



```
k2_labels = cluster.KMeans(n_clusters=10).fit_predict(X)
plt.scatter(s2_embedding[:, 0], s2_embedding[:, 1], c=k2_labels, s=0.1,
```



## Dimensional reduction

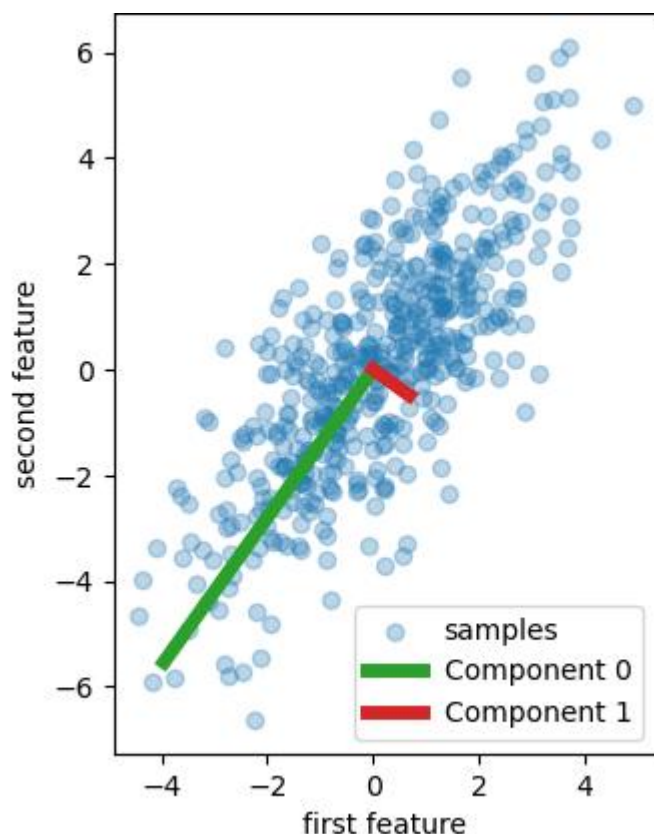
The dimensional reduction (DR) means that we need to find few vectors that describe well the high dimensional problem. This is not an easy task. One method is PCA (Principal component analysis) which basically create a new coordinate system of the data. The data varies most on the first component and less on the second, etc.

The length of these vectors will tell the variation. If they are similar the data is randomly distributed.

Now we can reprint the data with a few lowest vectors in this new coordinate system. We may find clusters better in this way.

The use of PCA is easy `pca = PCA(n_components=2)`

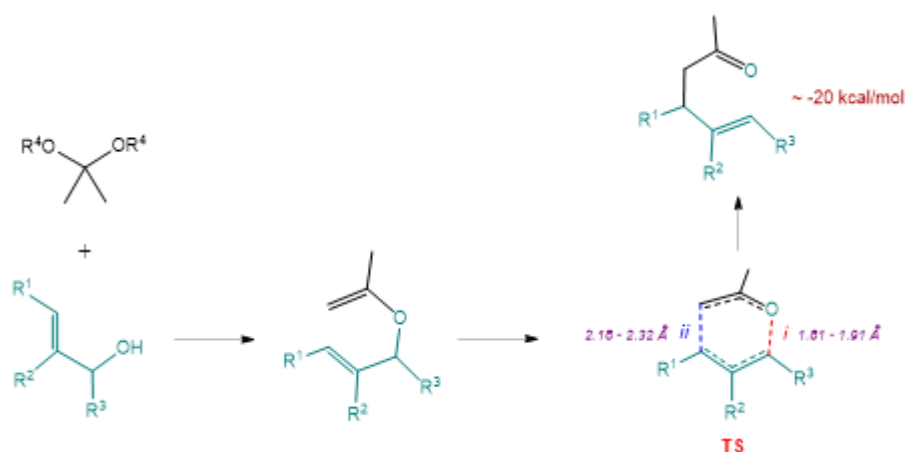
There are several other DR methods. One simple one is Singular Value Decomposition (SVD)



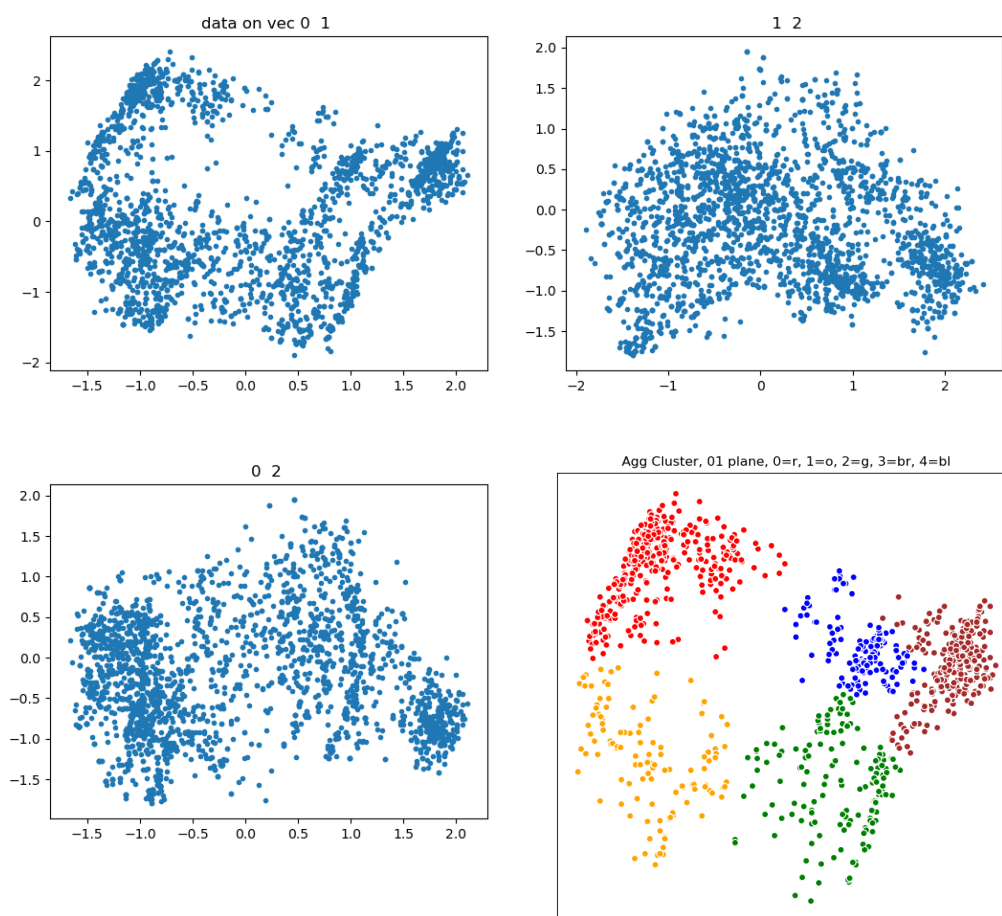
One need to keep in mind that the descriptors matters also in the unsupervised learning. We plot the data according to the descriptors.

### The Claisen project

We did a lot of ML analysis of the Claisen reaction. We do not have much experimental data but we had a lot of potential molecules (1500 in the first set and 2700 in the second). We tried to use the classification approach in this project. The molecules are described using the SMILES and then a Fingerprint analysis is done. (128 element vectors). The descriptor is the Fingerprint.



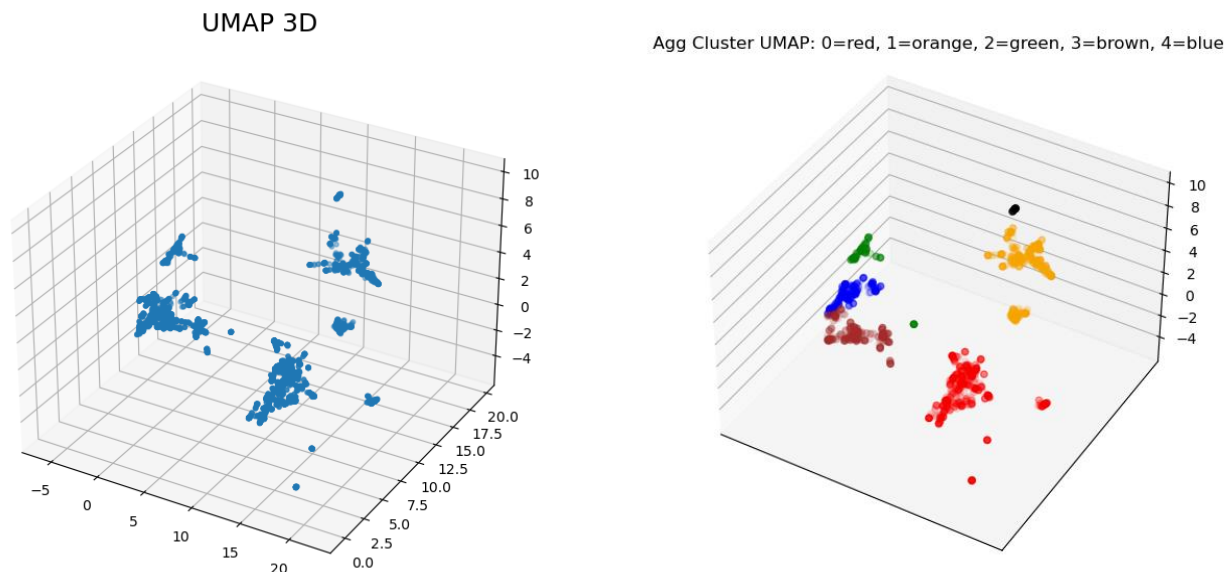
The PCA analysis is not very useful. Below are the projections of the 3 lowest PCA vectors. and in the corner the Agg Cluster analysis of (0,1) projection.



One of the most powerful DR method I have used is UMAP. It is not in sklearn but it can be loaded from net. It has very good web page: [Using UMAP for Clustering — umap 0.5 documentation \(umap-learn.readthedocs.io\)](https://umap-learn.readthedocs.io/en/latest/using-umap-for-clustering.html)

For the same data as before. On the left the raw data with 3 vectors and on the right with Agg cluster analysis.

Now the idea is to do experiments on different clusters and to see if they are chemically different. The preliminary data shows rather small differences. We have also done some DFT calculations of the barriers.



## Where we can get the data for ML projects

In chemical and material science problems we have some large experimental databases (DB), like the crystal structure DB's but for many properties we do not have large DB's. Individual values can be found from the literature but if we need thousands of numbers large scale DFT computations are a promising approach. The experimental data from various sources can contain errors whereas if the DFT computations are done systematically the data is of good quality. Of course, the DFT is not perfect but for ML we need trends and large data sets. This is the reason why most chemistry and materials science ML projects are based on DFT calculations.

Because the DFT results are so useful (for ML) there are also DB's for the DFT results, like NOMAD. A good review of the Databases is *Himanen et al. Adv. Sci.* **2019**, *6*, 1900808, DOI: 10.1002/adv.201900808

**NOMAD:** Provides storage for full input and output files of all important computational materials science codes, with multiple big-data services built on top. Contains over 50 236 539 total energy calculations.

Warning the databases are not always easy to use and the data quality can be quite poor. We did a M.Sc. study of chemical reactions using DFT DB's and the results were not very good. We are in the beginning of the DFT DB's and the rules of what one needs to store in the DB's does not exist. It also seems that the data in the DB's are not checked very carefully. I hope that the quality DB's will improve in the future. Naturally this criticism does not apply to all databases.

## High throughput computations

If one need to do 1000's of (DFT) computations the workflow need to be automatize. In simple cases this can be done with **Unix scripts** and in larger projects the are tools like FireWorks. Note that both the computations and the data analysis need to be automatized. On the scrips level this is not very difficult but to handle crashed or not converged jobs is not easy.

An example: Adding hydrogens on some surface, like carbon nanotube. Initial geometry is easy to make by adding H on top of an atom or between atoms (this is a bit more difficult). The DFT optimization is easy providing the system converges well. If the top site is not stable this will cause problems. Once the computations are done some analysis is needed, like some atom distances or HOMO, LUMO energies etc. A GPAW-type code that is partly written with python would be ideal but most quantum chemistry codes do not use python. The learning target is the H binding energy.

<https://materialsproject.github.io/fireworks/>

FireWorks is a free, [open-source](#) code for defining, managing, and executing workflows. Complex workflows can be defined using Python, JSON, or YAML, are stored using MongoDB, and can be monitored through a built-in web interface. Workflow execution can be automated over arbitrary computing resources, including those that have a queueing system.

## Data quality

**Remember: GARBAGE IN GARBAGE OUT**

**ALL INFORMATION IS IN THE DATA**

Data quality is essential to ML. Wherever you get the data one should be skeptical of its quality.

Are there some chemical bonds the data should fulfill?  
In large databases, is there bad data or missing data.

When the ML parity plot is done are there some outliers in the data. The outliers can be due to the poor ML model OR form poor input data.

When doing 1000's of DFT calculations, are all the results converged?  
How accurate the DFT is?

When using external DB's how do you know the data quality.

## Predictability

The predictability is one of the hardest questions in ML. We can easily analyse the predictability of the **data set we have** but what happen when we go outside the data. If the new molecules (or materials) are similar we can expect reasonable predictions. But what is "similar"?

The larger and more diverge the learning DB is the more we can predict. We need tools to analyse the divergence of the DB's then we can have some information on what can be predicted.

We need bravely test the predictions with new molecules. To test the predictions we need the correct answer. It is quite easy if you use computational data but much harder with experimental data.

