Software Usage Notice

- This software is provided for research and educational purposes only. Commercial use is not permitted.
- Do not redistribute this software, including modified versions, without prior permission from the author (aoyagi@st.seikei.ac.jp).
- The software is provided "as is," without warranty of any kind. Users are responsible for verifying the results and assessing their suitability for the intended purpose.
- If you encounter any problems or have questions, please contact aoyagi@st.seikei.ac.jp.

Software Download:

https://www.dropbox.com/scl/fo/27ben5iztho0p0dnqy91o/ALV61N3wQClZv0f8oMniuoE?rlkey=jel5kxn2r7pr8on0svep65801&st=gh20ua9h&dl=0

Unsupervised Learning Methods: PCA, MCR-ALS, and Sparse Autoencoder (SAE)

Unsupervised learning methods are useful for exploring the intrinsic structure of measurement data without requiring prior labels.

- **PCA** provides an overview of the dataset by reducing its dimensionality and revealing the major sources of variation.
- **MCR-ALS** decomposes the data into pure components (or groups of related variables) together with their corresponding concentration or score profiles.
- **Sparse Autoencoder (SAE)** extracts latent features using a nonlinear neural network. Because SAE can model nonlinear relationships, it may reveal finer structural information than MCR-ALS for some datasets.

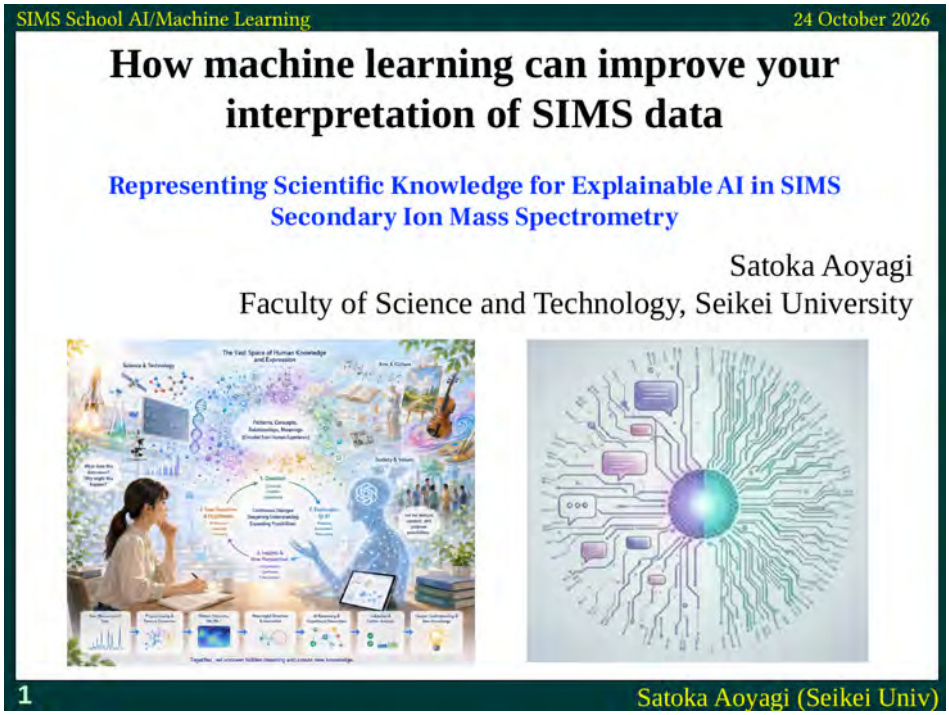
Although MCR-ALS and SAE often produce similar results, SAE may provide more detailed feature extraction when nonlinear effects are present.

Supervised Learning Methods: Random Forest (RF)

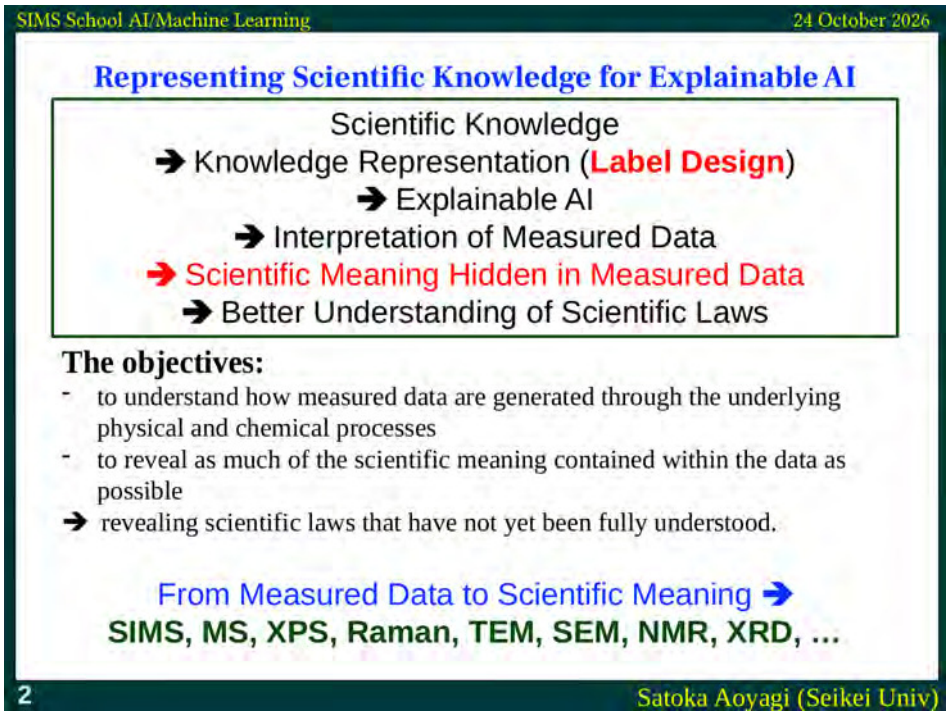
Supervised learning methods are useful for identifying variables related to a target peak, target image, or target variable that has been identified through exploratory analysis. Typical applications include

- identification of mass peaks related to a selected peak,
- discovery of variables exhibiting behavior similar to a selected variable,
- identification of mass images related to a selected target image.

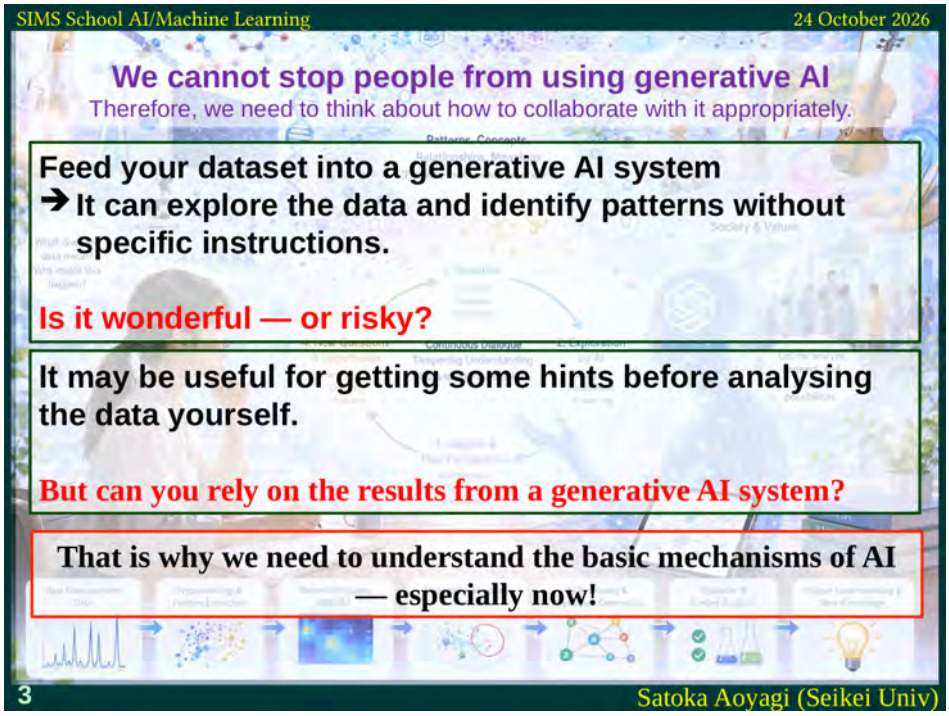
Random Forest (RF) is generally applicable to a wide variety of datasets because it can model nonlinear relationships among variables.



Today I will discuss how machine learning can improve our interpretation of SIMS data. I will begin with a short demonstration of generative AI, then focus on multivariate analysis and machine-learning methods for spectra and images. The aim is to connect the results with the physical and chemical processes that generate the measured data.



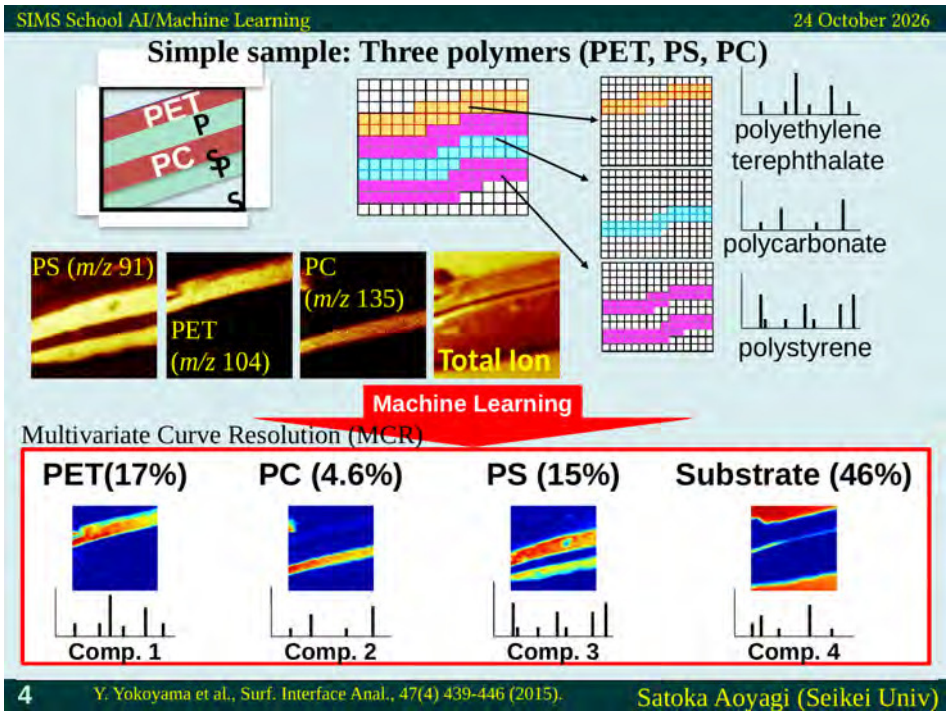
The objective is to uncover the scientific meaning in the measured data. Researchers first represent relevant scientific knowledge in a form that a model can use, for example through labels and descriptors. Explainable methods then help us examine the relationships between the model results and the measured variables. Scientific interpretation and validation help us connect these findings to the underlying physics and chemistry.



First, I would like to briefly discuss the current state of AI.

Generative AI has advanced rapidly, and many of you may already use it in your studies or research. However, applying generative AI directly to SIMS data is not yet common. Here, I would like to show what happens when we try.

It may be useful for getting some hints before analysing the data ourselves. However, we also need to consider whether we can rely on its interpretation. This is why understanding the basic mechanisms of AI is important.



This sample contains PET, PS and PC on a substrate. The total-ion and characteristic-peak images show their spatial distributions, while the MCR results provide component images and corresponding spectra. These results provide a reference for the generative-AI demonstration that follows. Chemical assignments still rely on interpreting the extracted spectra.

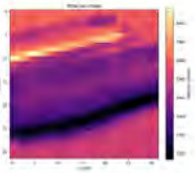
SIMS School AI/Machine Learning 24 October 2026

Generative AI

PET, PS, and PC

Polymer sample dataset

Mass (Da)	1.01	2.01	12.00	13.01	14.01	15.02	...	1119.72	1120.76	1163.75
1	1	0	0	0	1	8	...	0	0	0
2	5	0	0	0	0	9	...	0	0	0
3	0	0	0	0	2	13	...	0	0	0
4	1	0	0	0	0	6	...	0	0	0
5	2	0	1	0	0	8	...	0	0	0
6	0	0	0	0	0	7	...	0	0	0
7	0	0	0	0	1	4	...	0	0	0
8	0	0	0	0	0	5	...	0	0	0
9	2	0	0	0	0	10	...	0	0	0
10	2	0	0	0	1	5	...	0	0	0
11	0	0	0	0	0	7	...	0	0	0
12	0	0	0	0	0	8	...	0	0	0
13	0	0	1	1	1	7	...	0	0	0
14	2	0	0	0	0	10	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...
1023	1	0	0	0	2	6	...	0	0	0
1024	1	0	0	0	0	12	...	0	0	0



Total ion image (ToF-SIMS), 32 by 32 pixels, 1333 peaks (1~461 Da)

5 Satoka Aoyagi (Seikei Univ)

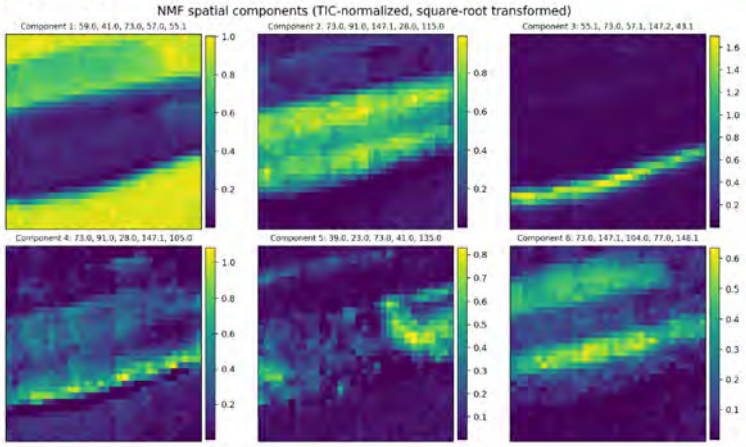
Suppose we have SIMS data from a sample with an unknown composition. In this demonstration, I knew the true composition, but I treated it as unknown.

First, I converted the SIMS data into text and uploaded it to generative AI systems, including ChatGPT and DeepSeek.

Here, the rows represent spatial pixels and the columns represent mass peaks. The dataset contains spectral and spatial information, but these numerical values alone do not explain the sample chemistry or measurement conditions.

SIMS School AI/Machine Learning 24 October 2026

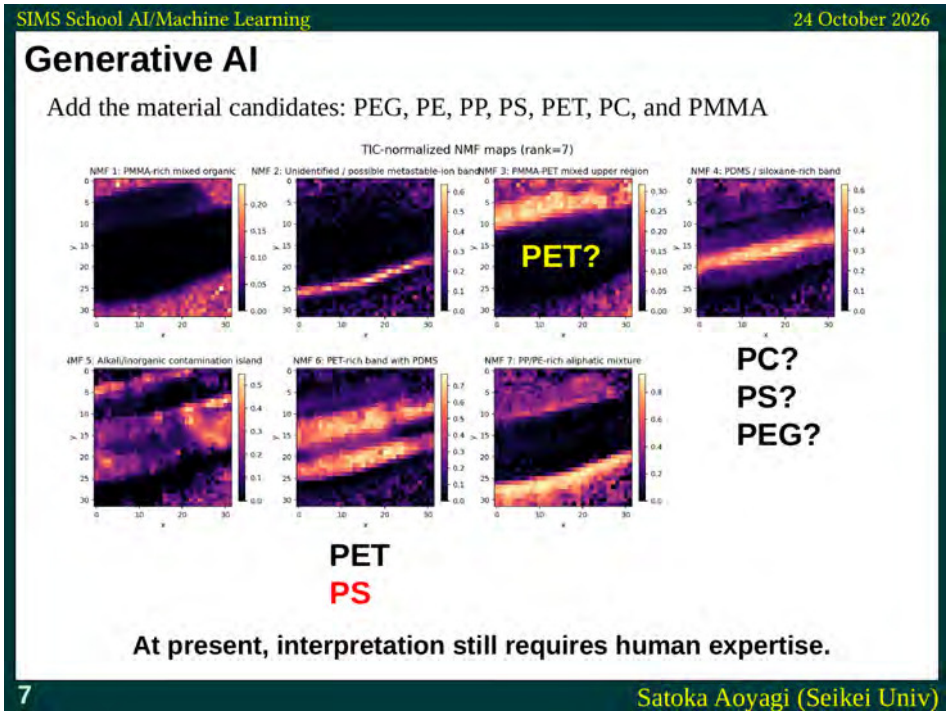
Generative AI



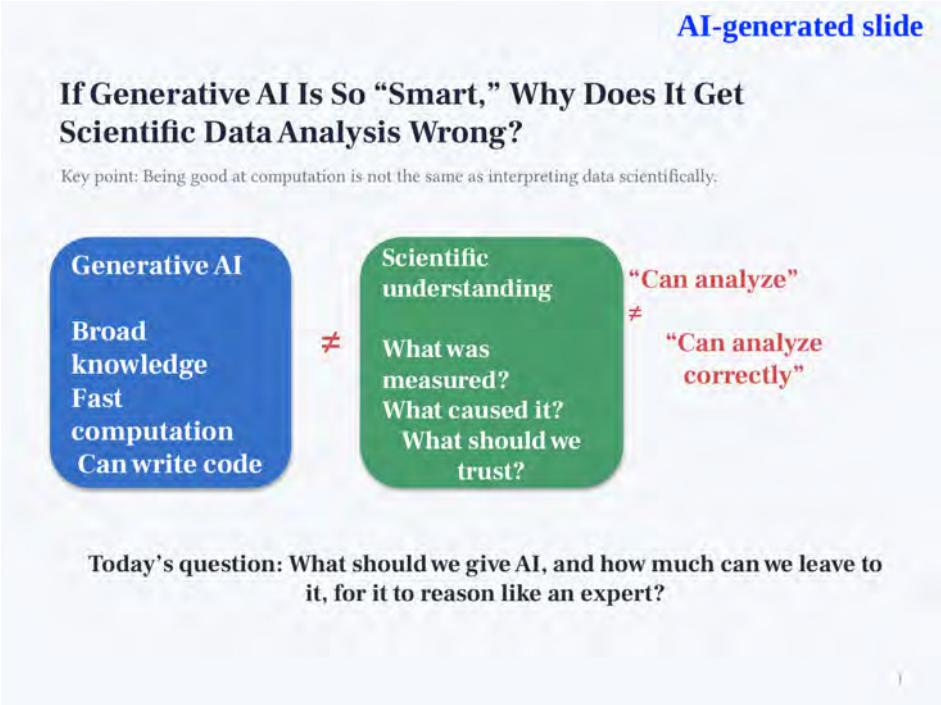
- Automatic analysis by this Generative AI did not identify main materials.
  - ➔ The identification based on fragment ions is not straightforward.
  - ➔ PDMS and Na/K contaminations were suggested.
- The main material distributions were indicated.

6 Satoka Aoyagi (Seikei Univ)

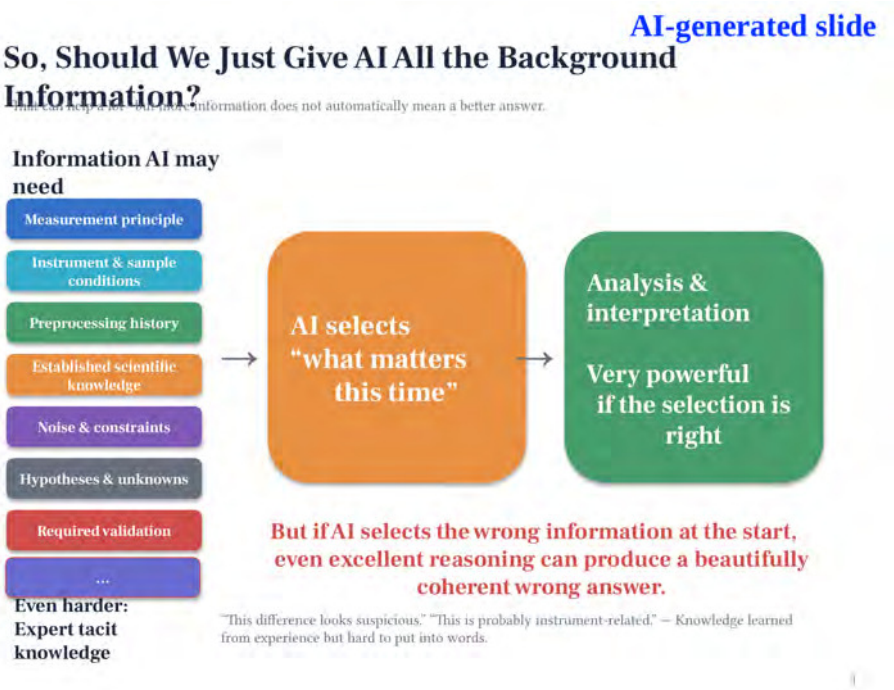
The automatic analysis indicated the main spatial distributions, but it did not identify the main polymers correctly. It suggested PDMS and sodium or potassium contamination instead. Separating regions and assigning their chemical identities are different tasks. The fragment-ion information alone did not provide enough context for reliable identification.



I then supplied a list of candidate materials: PEG, PE, PP, PS, PET, PC and PMMA. This was a candidate list, rather than a statement that all seven polymers were present. With these clues, the results from non-negative matrix factorization, NMF, and cluster analysis improved. Some assignments were correct. However, several regions remained uncertain, so interpretation still required human expertise.



The demonstration separates computational ability from scientific interpretation. The AI could process a matrix of mass peaks and spatial pixels, but I had not supplied the measurement conditions or relevant background knowledge about the sample. Without these clues, it could not reliably determine what the extracted patterns meant. This leads to the question on the next displayed slide: would supplying more information solve the problem?



Would providing enough information solve this problem? Not necessarily. Even when sufficient information is provided, important information may not be appropriately selected, connected, or used.

The challenge is therefore not only to provide sufficient information, but also to ensure that AI preserves and uses the knowledge essential to the analysis. This includes the measurement principles, conditions, preprocessing history, uncertainties and purpose of the analysis.

We usually express this knowledge through natural language, equations or structured descriptions. However, practical experience and scientific judgment can be difficult to express explicitly. How can we make that knowledge accessible to AI? This remains a major challenge.

SIMS School AI/Machine Learning 24 October 2026

AI Applications in Spectrum and Image Analysis

Purpose	Current Methods / Applications	Future
Dataset overview	PCA	–
Component extraction	MCR / NMF, SAE, SOM, ...	–
Variable interpretation	RF, LASSO, Shallow ANN, ...	–
Material prediction	Label design + RF/ANN	Prediction of unknown materials
Quantitative analysis	ANN	Matrix-effect correction
Noise reduction	ANN, HNN	Automatic denoising and preprocessing
Knowledge discovery	Hidden relationships among variables, important variables	Discovery of new scientific insights
Measurement optimization	–	Automatic optimization of acquisition conditions
Autonomous analysis	–	Measurement → Analysis → Re-measurement
Anomaly detection	–	Detection of unknown or unexpected samples

- How AI reaches its conclusions
- Scientific knowledge that reflect the physical and chemical processes underlying the measured data ➔ Scientific discoveries

10 Satoka Aoyagi (Seikei Univ)

Before discussing advanced machine learning applications, let us first look at the current machine learning methods that help us interpret complex SIMS datasets. This table summarizes both current and future applications of machine learning in SIMS and related analytical techniques.

Conventional multivariate analysis, such as principal component analysis, is still a powerful tool for interpreting complex dataset, particularly for obtaining an overview of the data. For extracting pure components or important features, unsupervised learning methods, such as multivariate curve resolution (MCR), a type of non-negativity matrix factorization (NMF) and autoencoders are particularly useful.

After unsupervised learning identifies characteristic components or spectral patterns, supervised learning methods can identify the variables that are most strongly associated with these findings.

More advanced applications include material prediction, quantitative analysis, noise reduction and knowledge discovery. Although, automatic denoising is still challenging today, it is likely to become increasingly practical in the coming years. Moreover, autonomous measurement systems may become feasible in the near future. Such systems could automatically optimize measurement conditions, search for the most informative regions for analysis, and even suggest the next sample preparation or measurement strategy.

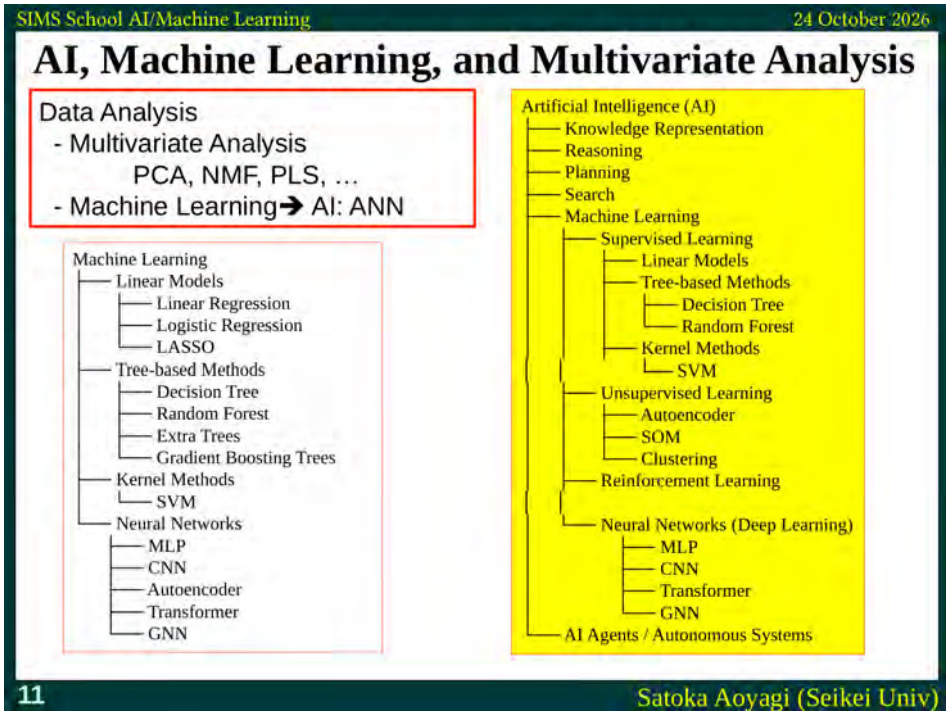
Although this future may sound very convenient, it also places greater responsibility on us as scientists. We must understand how AI reaches its conclusions rather than simply accepting its answers. At the same time, we need to provide AI with scientific knowledge that reflect the physical and chemical processes underlying the measured data. Collaboration between human scientific understanding and AI will be essential for achieving reliable and explainable scientific discoveries.

--

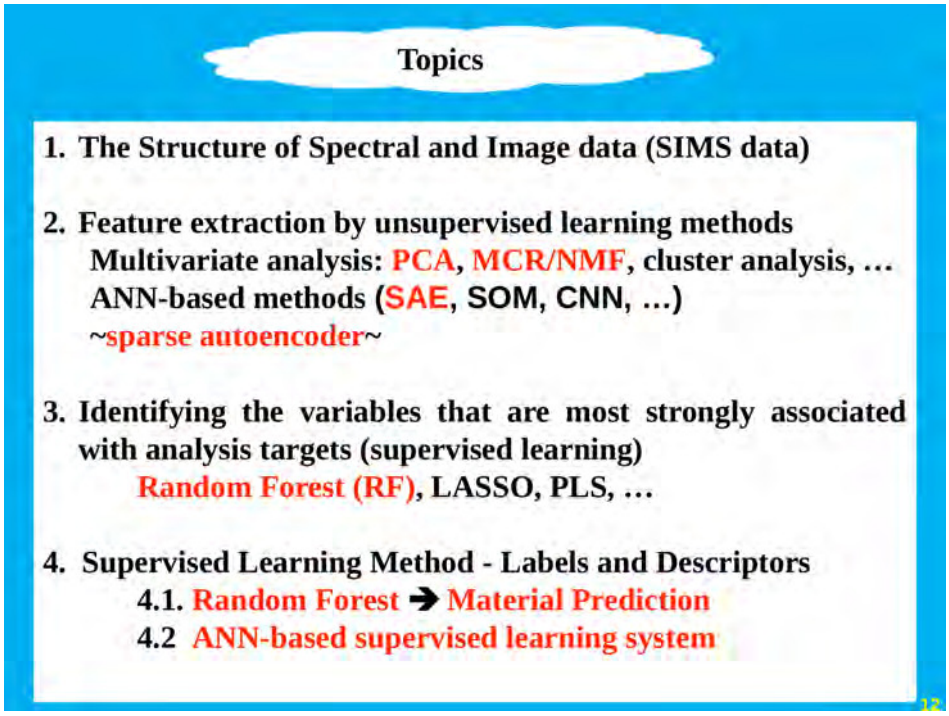
PCA provides an overview, while MCR, NMF and SAE extract component patterns or features.

Supervised methods can investigate variables related to a target, predict material-related labels or estimate composition.

The future column includes measurement optimization and autonomous analysis as possibilities, rather than capabilities demonstrated in this tutorial. Our focus is on understanding how the available methods support spectral and image interpretation.

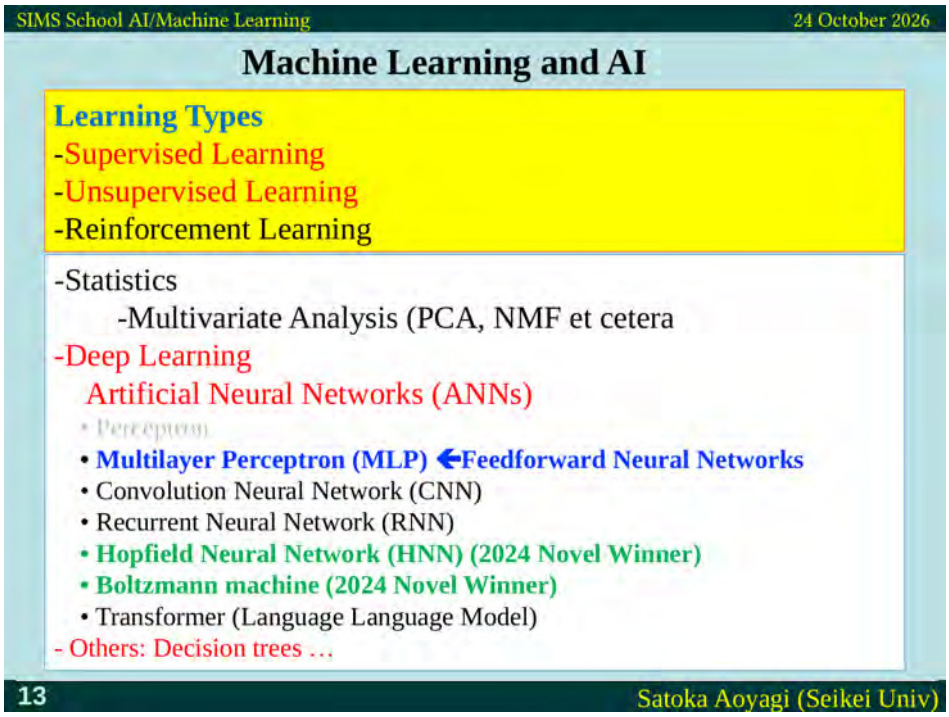


This optional overview places multivariate analysis and machine learning within the broader field of AI. It groups learning methods by their model families and learning tasks. The tutorial concentrates on PCA, MCR, SAE, Random Forest and shallow neural networks because their results can be examined in relation to measured spectra and images.

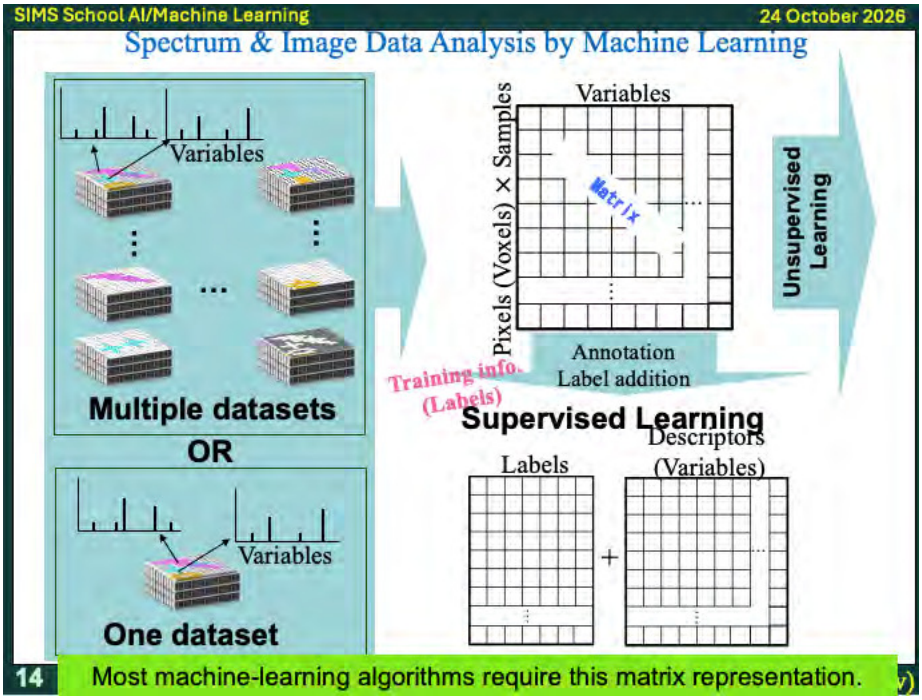


The tutorial begins with the matrix representation of spectral and image data. We then examine unsupervised feature extraction using PCA, MCR or NMF, and sparse autoencoders. Next, supervised learning helps us investigate variables associated with a target.

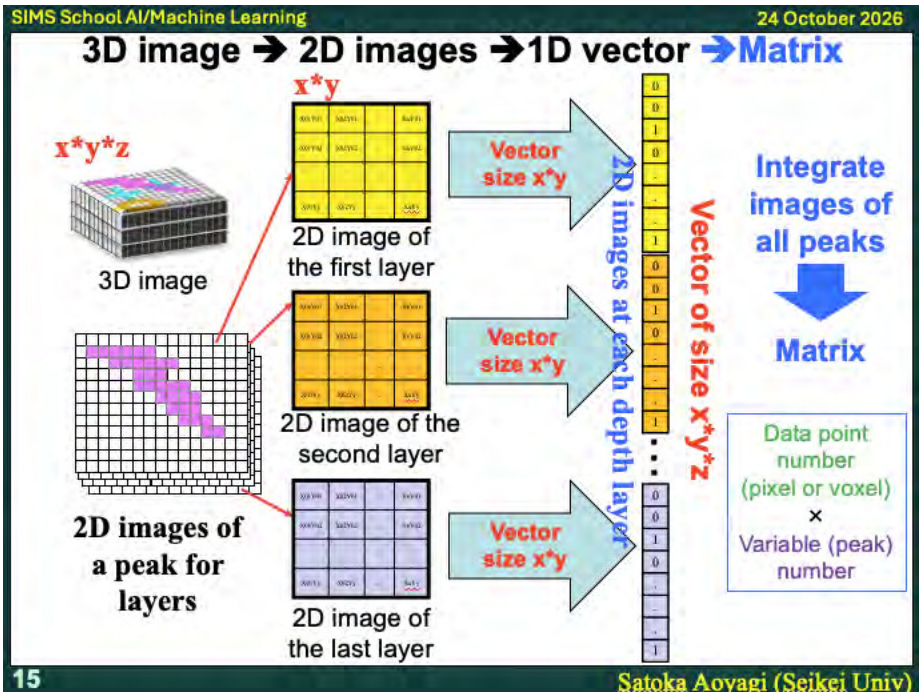
Finally, we introduce label and descriptor design to connect random forest material prediction and ANN-based quantitative analysis with scientific knowledge.



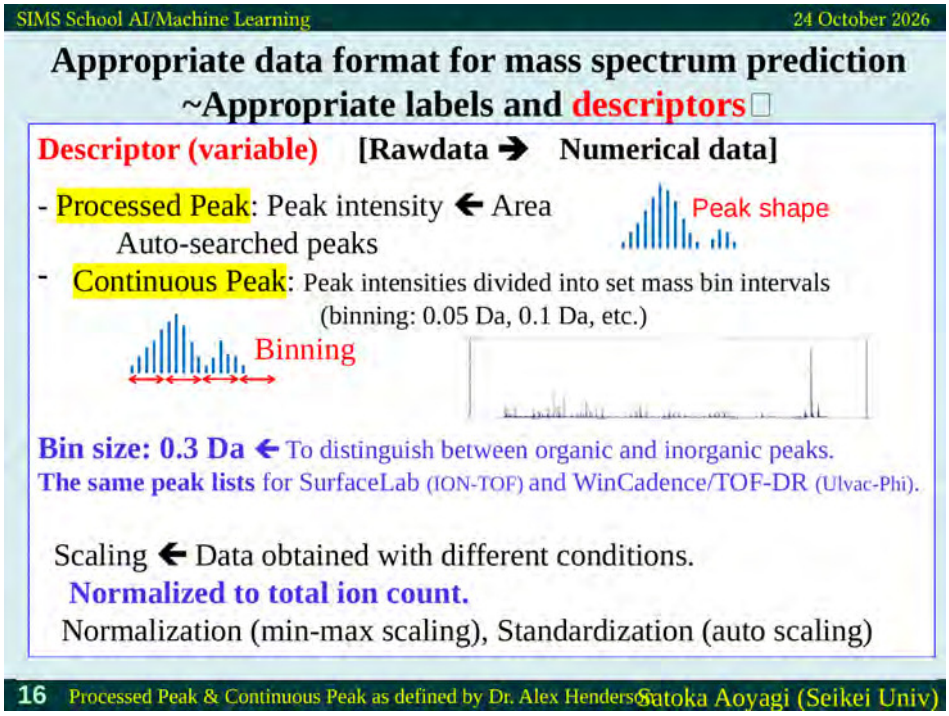
Machine-learning tasks include supervised, unsupervised and reinforcement learning. This tutorial focuses on the first two. The slide also lists model families, including conventional statistical methods, decision trees and several neural-network architectures. We will choose methods according to the analysis purpose and the information available for interpreting their outputs.



To apply machine learning to SIMS or MS imaging datasets, we first convert the data into a matrix because most algorithms require this representation. Each mass-peak image can be converted into a vector, and these vectors can then be combined to form a matrix. Unsupervised learning requires only the dataset itself, whereas supervised learning requires both the dataset and labels. The success of supervised learning often depends more on appropriate label design than on the choice of algorithm.



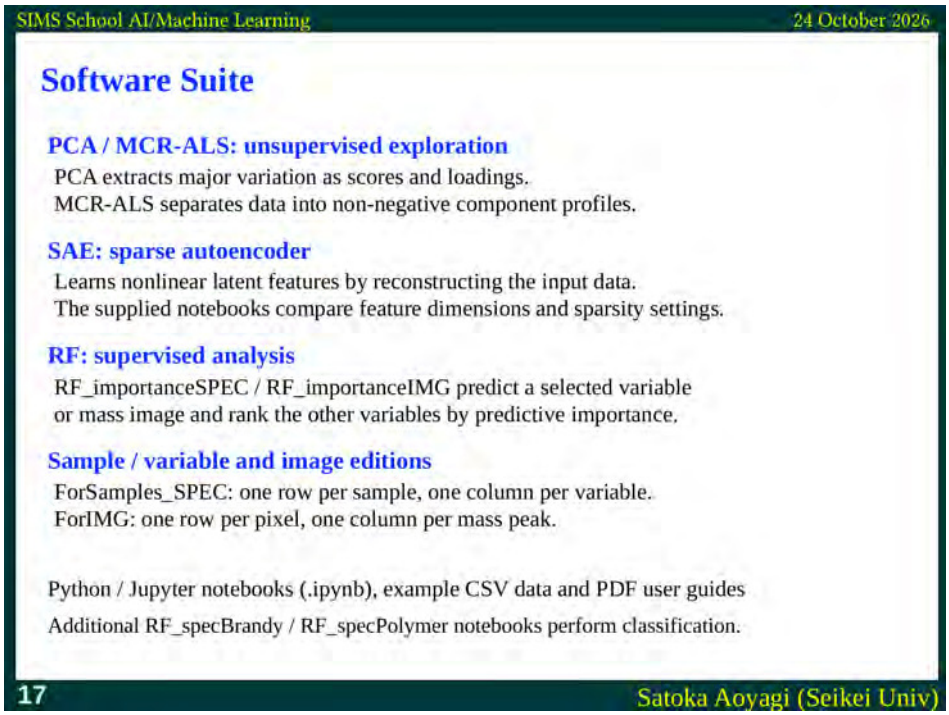
For a three-dimensional dataset, flatten each depth-layer image into a vector, and join the vectors end to end in a fixed order. Repeat that ordering for every mass peak to form the numerical matrix. After analysis, the same ordering lets us reshape scores or encoded values back into spatial images.



The descriptors are the numerical variables supplied to the model. They can be peak areas from peak picking or intensities integrated over fixed mass bins. A common peak list helps align spectra from different measurements.

Furthermore, peak picking is important for preparing appropriate descriptors for MS data. We can use the SIMS software provided by vendors, as well as the free software available on GitHub.

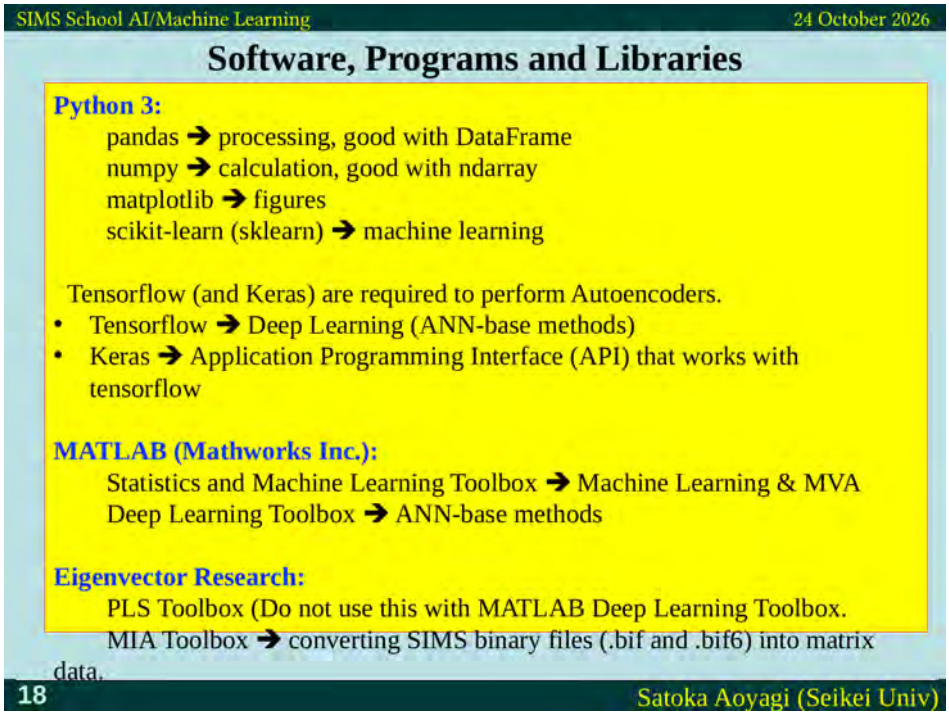
In addition, normalization and scaling change the relative contributions of variables, in this case mass peaks, so their choice should preserve the spectral information needed for the scientific question.



I've shared the software suite for this tutorial. The practical suite contains PCA, MCR-ALS and sparse-autoencoder notebooks for unsupervised learning analysis, and Random Forest notebooks for supervised analysis. The RF importance notebooks predict a selected numerical variable or mass image from the other variables. The additional RF_specBrandy and RF_specPolymer notebooks perform classification. There are separate editions for sample-by-variable data and pixel-by-mass data, with example CSV files and PDF user guides.

Sources: supplied SoftwareSuite/UnsupervisedLearning and SupervisedLearning notebooks and PDF user guides. PCA_sampleData_20260707.ipynb, MCRALS_sampleData_20260707.ipynb, SAE_sampleData_20260707.ipynb; PCA_20260615.ipynb, MCRALS_20260617.ipynb,

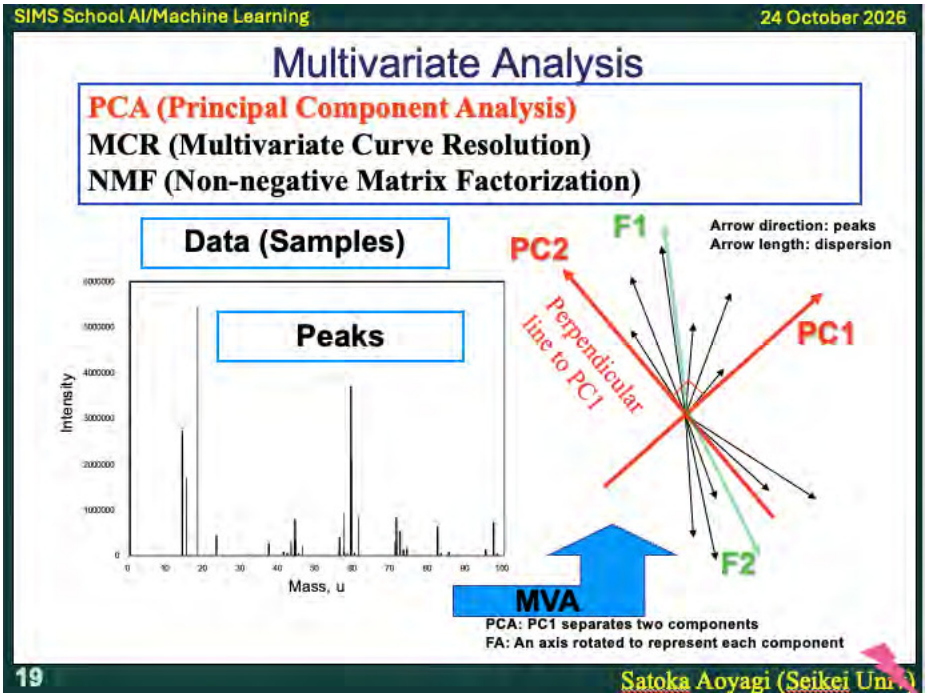
SAE_ParameterComparison_20260615.ipynb, Bayesian_SAE_20260615_v2.ipynb; RF_importanceSPEC_20260707.ipynb, RF_importanceIMG_20260707.ipynb, RF_specBrandy_20250803.ipynb, RF_specPolymer_20250803.ipynb.



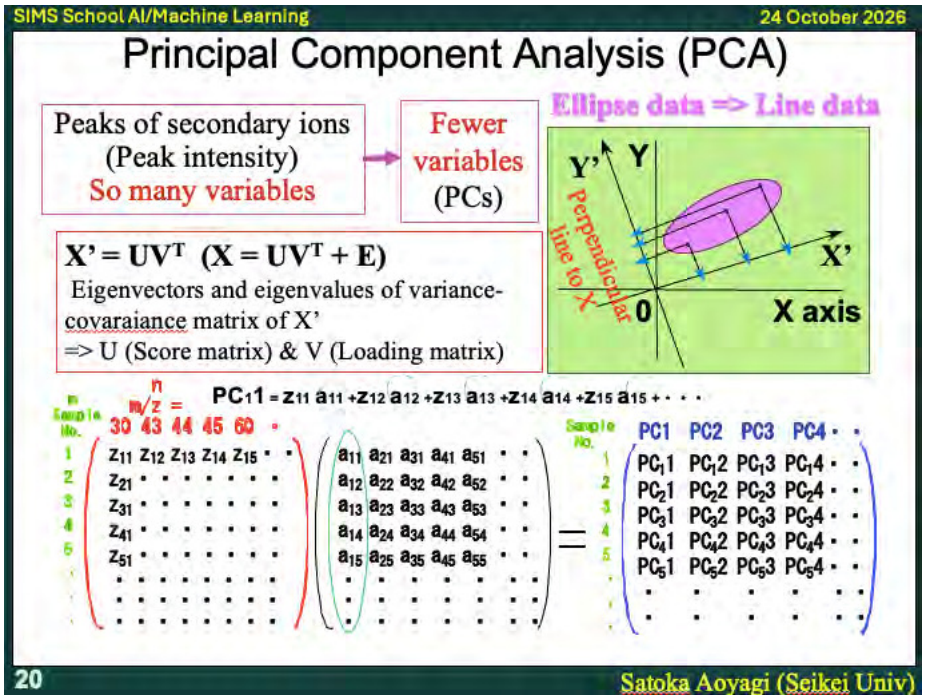
In Python, pandas is useful for handling tabular data, NumPy for numerical calculations, matplotlib for figures, and scikit-learn for many machine-learning methods. TensorFlow and Keras are used for the autoencoder examples.

MATLAB also provides toolboxes for machine learning and neural networks.

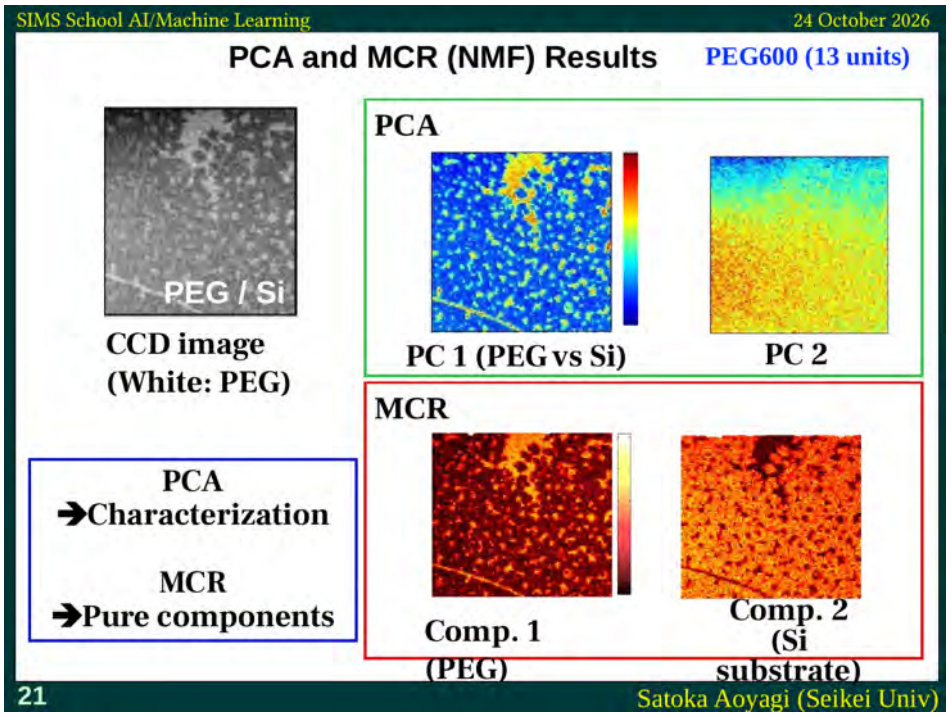
For SIMS image data, converting the original files into a numerical matrix is an important first step, and the MIA Toolbox is one option shown here.



PCA and MCR or NMF provide different ways to summarize a complex dataset. PCA finds directions that capture the greatest variation in the data. MCR or NMF aims to represent the data in terms of component profiles, subject to the selected constraints. The diagram illustrates why an axis that separates samples is not necessarily the same as an axis representing an individual material. We should interpret the results according to the purpose and assumptions of each method.

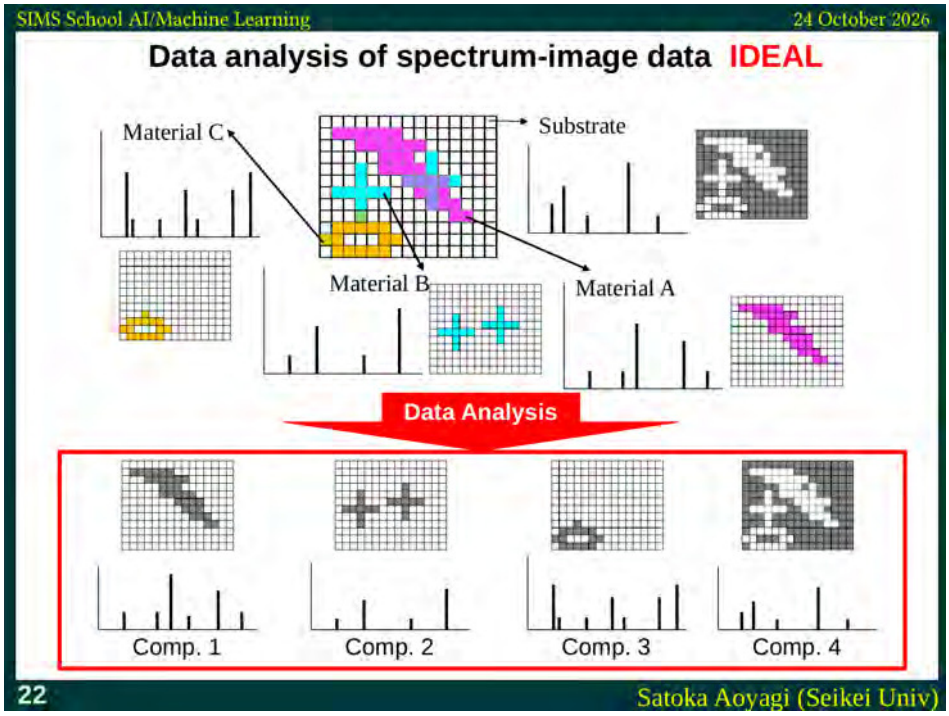


PCA replaces a large number of original variables with a smaller number of principal components. Each principal component is a linear combination of the original variables. The scores describe the samples or pixels in this new coordinate system, while the loadings show which mass peaks contribute to each direction. Together, scores and loadings give an overview of the dataset and help us interpret the differences among samples or regions.

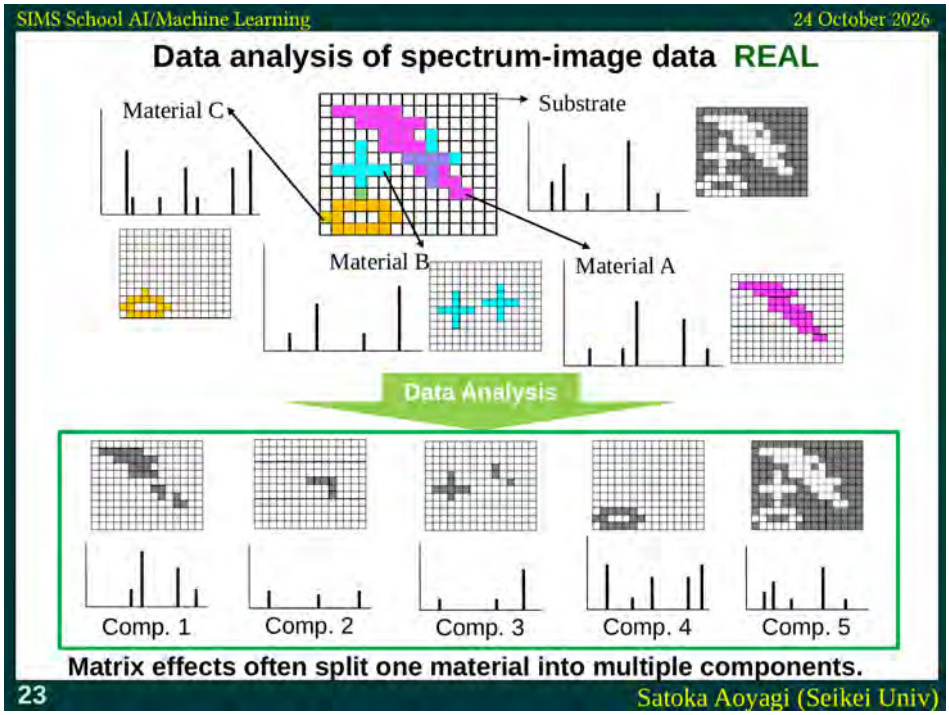


In this case, um, the pig sample measured with SIMS was analyzed using PCA and MCR.

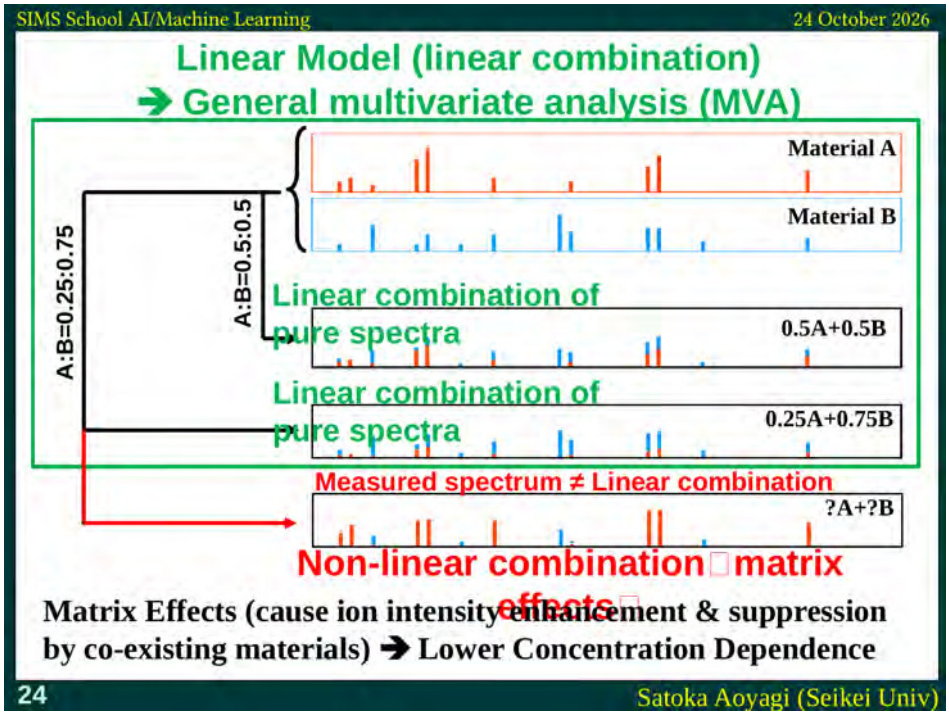
The first PCA component distinguishes PEG from the silicon substrate. The MCR result presents component images associated with PEG and the substrate. Comparing these images with the optical image helps us understand the spatial correspondence. PCA is useful for characterizing variation, while MCR can provide component-like profiles that are easier to connect with the materials present.



Unsupervised learning can extract important features and, under suitable conditions, profiles corresponding to pure components, together with associated images and spectra.

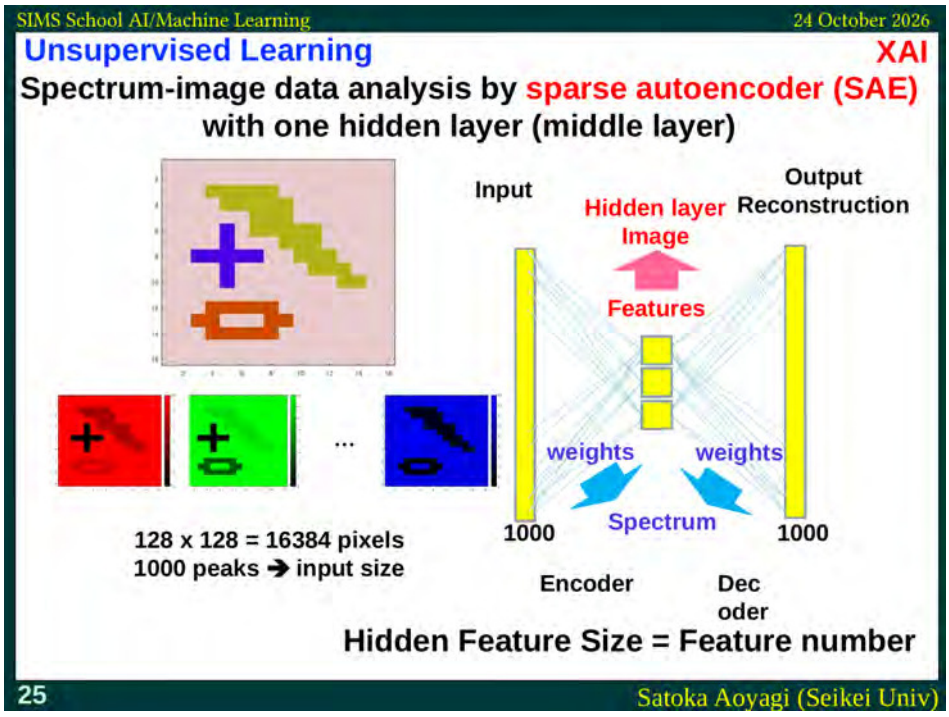


However, in real measurements, nonlinear factors such as matrix effects can distort the results of machine learning analysis.



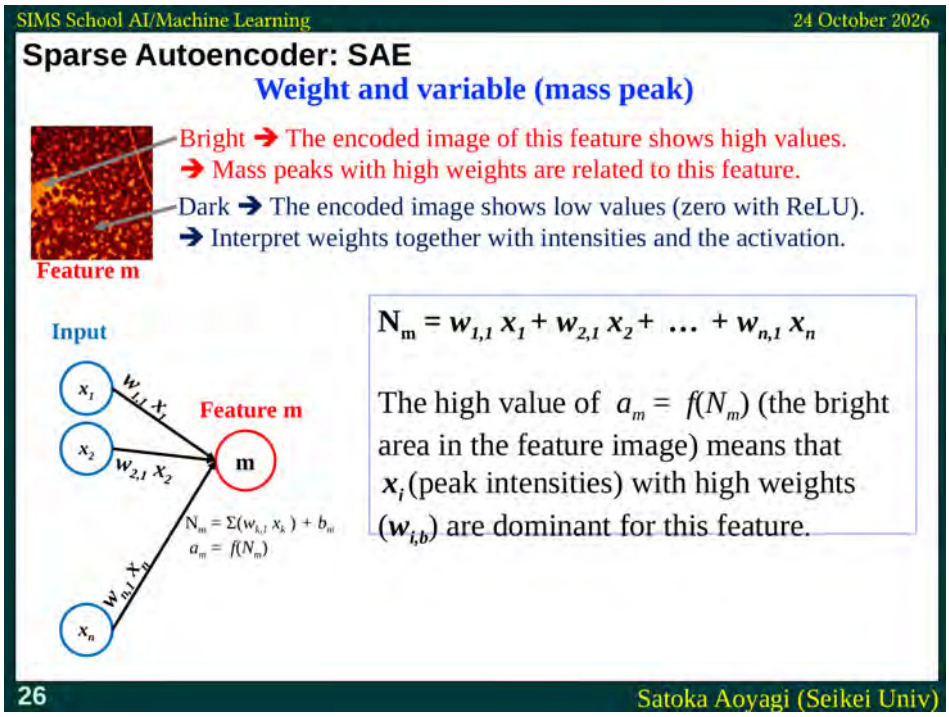
A linear mixture model represents the measured spectrum as a weighted combination of pure-material spectra. Coexisting materials can enhance or suppress ion intensities, causing the measured mixture to depart from that expectation.

For example, neural-network models can represent nonlinear relationships.



This is an example of an ANN-based method called an autoencoder. An autoencoder reconstructs the input data at the output layer. Through this process, important features are extracted in one or more hidden layers.

When we input SIMS imaging data, we can display the hidden-layer feature values for each pixel as images. In addition, we can examine the weight matrices connecting the input, hidden, and output layers. These weights can provide clues about how variables, such as mass peaks, relate to each extracted feature.



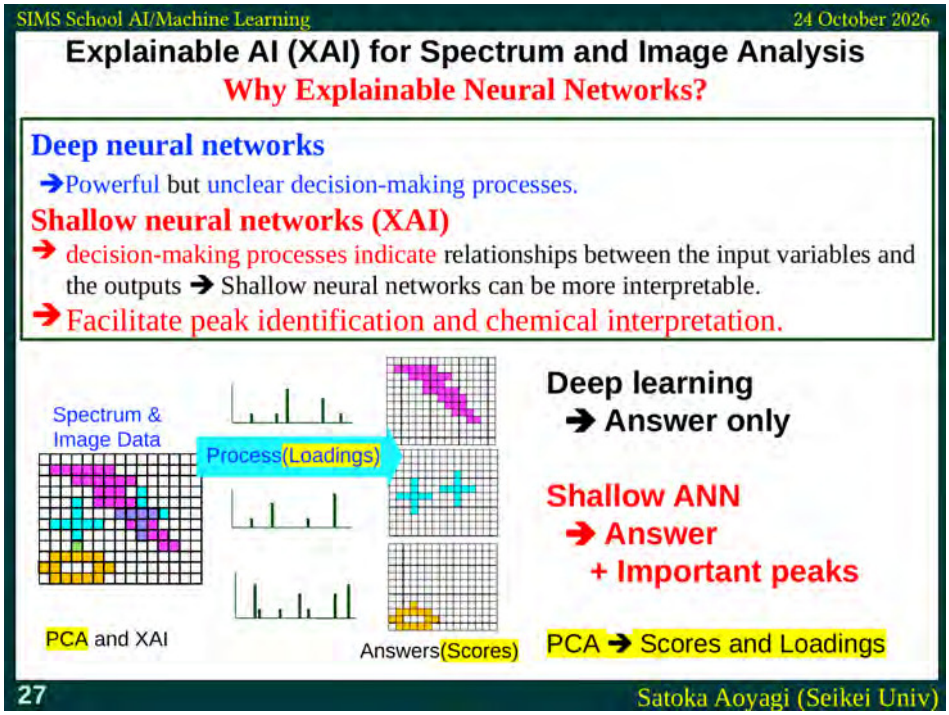
When SIMS imaging data is fed into a sparse autoencoder, the values at each hidden-layer node can be displayed as an image, similar to a PCA score image. The weights between layers can provide clues about which mass peaks contribute to each feature, somewhat like PCA loadings.

For each hidden node, the network calculates a weighted sum of the input intensities, adds a bias, and applies an activation function.

Bright pixels indicate higher encoded values, while dark pixels indicate lower values, including zero. With ReLU, these encoded values are non-negative.

Bright pixels may reflect contributions from mass peaks with high intensities and large positive encoder weights.

(Mass peaks with large positive decoder weights can help us interpret the spectrum associated with each extracted feature.)



For chemical interpretation, we need to examine the variables associated with an output, as well as the output itself. PCA provides scores and loadings. A shallow neural network can likewise expose connections between the input peaks and hidden features or output labels. This makes the relationships easier to inspect, although the weights still require interpretation in the context of the activation functions and measured spectra.

CSV files supplied with the suite

PCA / MCR / SAE example: poly_mass999_pix032.csv
1,024 pixels = 32 × 32, with 999 mass variables.
First row: Pixel and mass values. First column: pixel numbers.

Reading settings for this example
data_start_row = 1 data_start_col = 1
use_mass_axis = True use_pixel_axis = True
image_size_mode = "auto" (32 × 32 for this file)

Relation to the PEG example: poly_mass570_pix128.csv
The preceding PEG slide describes 128 × 128 pixels and 570 peaks.
Use the actual dimensions and mass list of each supplied CSV.

Brandy example (already autoscaled): brandy_singleLabel.csv
brandy_singleLabel.csv contains 24 samples × 15 aroma variables.
Each variable has mean ≈ 0 and population standard deviation ≈ 1.
Use scaling_method = "none" for PCA to avoid repeating autoscaling.

Now, I'd like to demonstrate how to use the software I've shared with you.

The supplied imaging CSV differs in format and size from the polymer dataset used in the generative AI demonstration. The shared files also include a brandy dataset, which contains spectral information but no images.

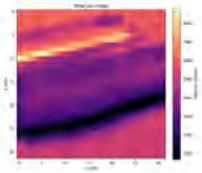
Skip the header row and Pixel column when extracting the numerical matrix. The alternative data_mass.csv has no Pixel column, so its column settings differ. Check the actual dimensions and header layout of the file used for each run.

Sources: supplied SoftwareSuite notebooks, example CSV files and 2026 user guides.

poly_mass999_pix032.csv

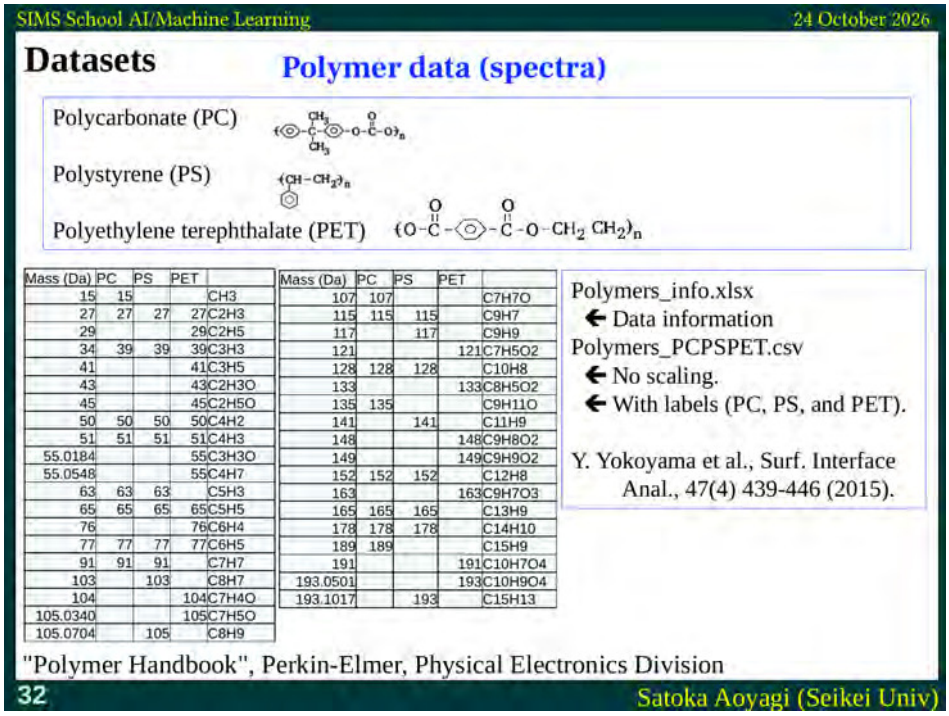
PET, PS, and PC

Mass (Da)	1.01	2.01	12.00	13.01	14.01	15.02	...	1119.72	1120.76	1163.75
1	1	0	0	0	1	8	...	0	0	0
2	5	0	0	0	0	9	...	0	0	0
3	0	0	0	0	2	13	...	0	0	0
4	1	0	0	0	0	6	...	0	0	0
5	2	0	1	0	0	8	...	0	0	0
6	0	0	0	0	0	7	...	0	0	0
7	0	0	0	0	1	4	...	0	0	0
8	0	0	0	0	0	5	...	0	0	0
9	2	0	0	0	0	10	...	0	0	0
10	2	0	0	0	1	5	...	0	0	0
11	0	0	0	0	0	7	...	0	0	0
12	0	0	0	0	0	8	...	0	0	0
13	0	0	1	1	1	7	...	0	0	0
14	2	0	0	0	0	10	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...
1023	1	0	0	0	2	6	...	0	0	0
1024	1	0	0	0	0	12	...	0	0	0



Total ion image (ToF-SIMS), 32 by 32 pixels, 999 peaks (*m/z* 1~461)

Here is an overview of the polymer imaging dataset.



Mass (Da)	PC	PS	PET	
15	15			CH3
27	27	27	27	C2H3
29			29	C2H5
34	39	39	39	C3H3
41			41	C3H5
43			43	C2H3O
45			45	C2H5O
50	50	50	50	C4H2
51	51	51	51	C4H3
55.0184			55	C3H3O
55.0548			55	C4H7
63	63	63		C5H3
65	65	65	65	C5H5
76			76	C6H4
77	77	77	77	C6H5
91	91	91		C7H7
103		103		C8H7
104			104	C7H4O
105.0340			105	C7H5O
105.0704		105		C8H9

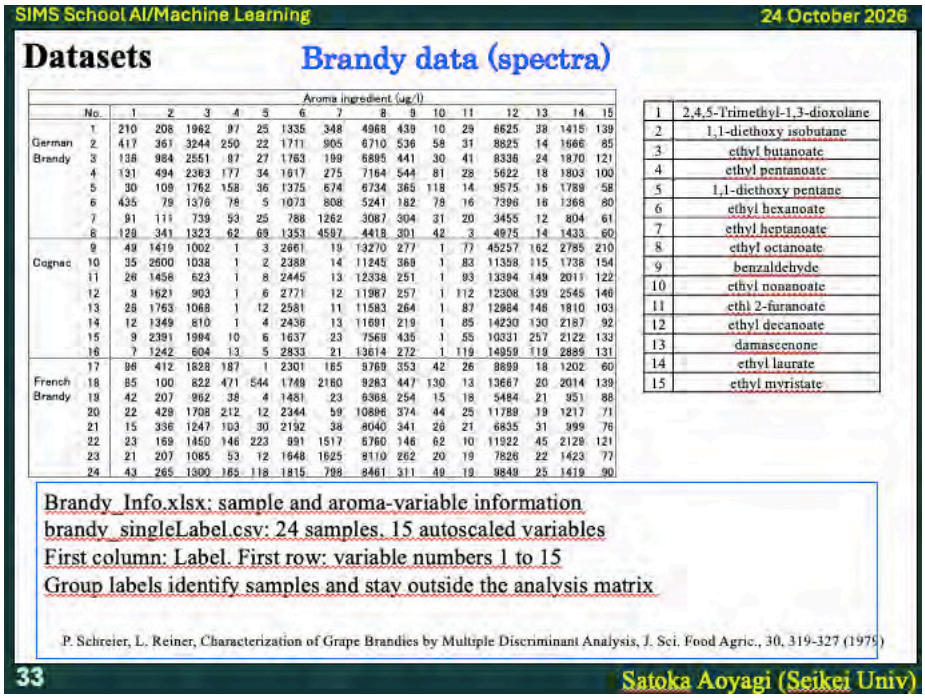
Mass (Da)	PC	PS	PET	
107	107			C7H7O
115	115	115		C9H7
117		117		C9H9
121			121	C7H5O2
128	128	128		C10H8
133			133	C8H5O2
135	135			C9H11O
141		141		C11H9
148			148	C9H8O2
149			149	C9H9O2
152	152	152		C12H8
163			163	C9H7O3
165	165	165		C13H9
178	178	178		C14H10
189	189			C15H9
191			191	C10H7O4
193.0501			193	C10H9O4
193.1017		193		C15H13

This slide provides information about the polymers in this dataset.

This dataset contains spectra of PC, PS, and PET. The table lists characteristic fragment ions and also shows that some nominal masses can correspond to different ion formulas.

For supervised learning, the material identities are supplied as labels.

These examples illustrate why both the mass resolution and the chemical interpretation of the descriptors matter.



No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	210	208	1962	97	25	1335	348	4968	439	10	29	8625	38	1415	139
German Brandy 2	417	361	3244	250	22	1711	905	6710	536	58	31	8825	14	1666	85
German Brandy 3	138	984	2551	87	27	1763	199	6895	441	30	41	8336	24	1870	121
German Brandy 4	131	494	2363	177	34	1617	275	7164	544	81	28	5622	18	1803	100
German Brandy 5	30	109	1762	158	36	1375	674	6734	365	118	14	9575	16	1789	58
German Brandy 6	435	79	1376	78	5	1073	808	5241	182	79	16	7396	16	1368	80
German Brandy 7	91	111	739	53	25	788	1262	3087	304	31	20	3455	12	804	61
German Brandy 8	129	341	1323	62	69	1353	4597	4418	301	42	3	4975	14	1433	60
Cognac 9	48	1419	1002	1	3	2661	19	13270	277	1	77	45257	162	2785	210
Cognac 10	35	2600	1038	1	2	2388	14	11245	368	1	83	11358	115	1738	154
Cognac 11	26	1456	623	1	8	2445	13	12338	251	1	93	13394	149	2011	122
Cognac 12	9	1621	903	1	6	2771	12	11987	257	1	112	12308	139	2545	146
Cognac 13	28	1763	1068	1	12	2581	11	11583	264	1	87	12984	148	1810	103
Cognac 14	12	1349	810	1	4	2438	13	11691	219	1	85	14230	130	2187	92
Cognac 15	9	2391	1994	10	6	1637	23	7569	435	1	55	10331	257	2122	133
Cognac 16	7	1242	604	13	5	2833	21	13614	272	1	119	14959	119	2889	131
French Brandy 17	96	412	1828	187	1	2301	185	9769	353	42	26	8899	18	1202	60
French Brandy 18	85	100	822	471	544	1749	2180	9283	447	130	13	13667	20	2014	139
French Brandy 19	42	207	962	38	4	1481	23	6368	254	15	18	5484	21	951	88
French Brandy 20	22	429	1708	212	12	2344	59	10896	374	44	25	11789	19	1217	71
French Brandy 21	15	336	1247	103	30	2192	38	8040	341	26	21	6835	31	999	76
French Brandy 22	23	169	1450	146	223	991	1517	5760	146	62	10	11922	45	2129	121
French Brandy 23	21	207	1085	53	12	1648	1625	8110	262	20	19	7826	22	1423	77
French Brandy 24	43	265	1300	165	118	1815	798	8461	311	49	19	9849	25	1419	90

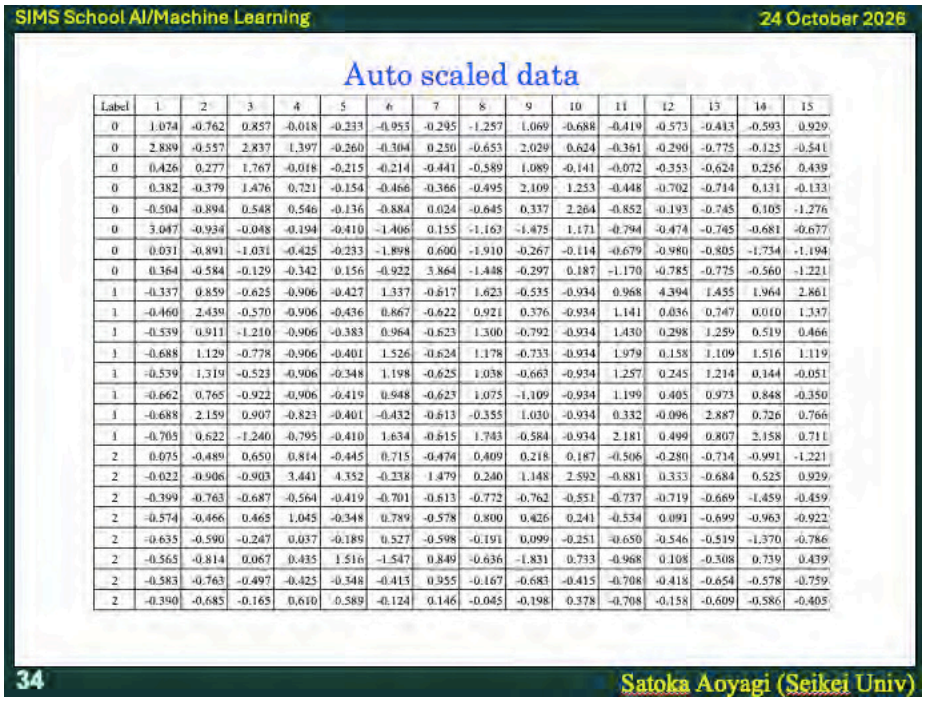
Next, here is an overview of the brandy dataset. This dataset has already been autoscaled.

The Brandy example contains 24 samples and 15 aroma-compound variables.

The groups include Cognac, French brandy and German brandy.

Brandy_Info.xlsx describes the samples and variables, and brandy_singleLabel.csv contains the autoscaled measurements. Its first row identifies the variables, and its first column contains group labels that must stay outside the numerical matrix.

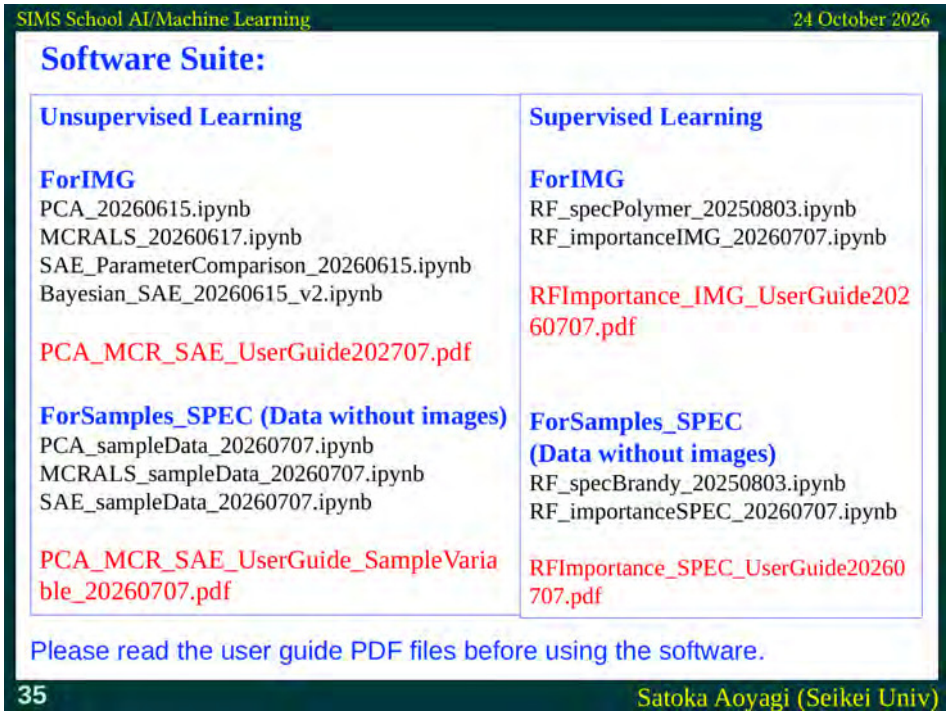
This small dataset lets us examine individual compounds alongside the extracted features.



Label	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1.074	-0.762	0.857	-0.018	-0.233	-0.953	-0.295	-1.257	1.069	-0.688	-0.419	-0.573	-0.413	-0.593	0.929
0	2.889	-0.557	2.837	1.397	-0.260	-0.304	0.250	-0.653	2.029	0.624	-0.361	-0.290	-0.775	-0.125	-0.541
0	0.426	0.277	1.767	-0.018	-0.215	-0.214	-0.441	-0.589	1.089	-0.141	-0.072	-0.353	-0.624	0.256	0.439
0	0.382	-0.379	1.476	0.721	-0.154	-0.466	-0.366	-0.495	2.109	1.253	-0.448	-0.702	-0.714	0.131	-0.133
0	-0.504	-0.894	0.548	0.546	-0.136	-0.884	0.024	-0.645	0.337	2.264	-0.852	-0.193	-0.745	0.105	-1.276
0	3.047	-0.934	-0.048	-0.194	-0.410	-1.406	0.155	-1.163	-1.475	1.171	-0.794	-0.474	-0.745	-0.681	-0.677
0	0.031	-0.891	-1.031	-0.425	-0.233	-1.898	0.600	-1.910	-0.267	-0.114	-0.679	-0.980	-0.805	-1.734	-1.194
0	0.364	-0.584	-0.129	-0.342	0.156	-0.922	3.864	-1.448	-0.297	0.187	-1.170	-0.785	-0.775	-0.560	-1.221
1	-0.337	0.859	-0.625	-0.906	-0.427	1.337	-0.617	1.623	-0.535	-0.934	0.968	4.394	1.455	1.964	2.861
1	-0.460	2.439	-0.570	-0.906	-0.436	0.867	-0.622	0.921	0.376	-0.934	1.141	0.036	0.747	0.010	1.337
1	-0.539	0.911	-1.210	-0.906	-0.383	0.964	-0.623	1.300	-0.792	-0.934	1.430	0.298	1.259	0.519	0.466
1	-0.688	1.129	-0.778	-0.906	-0.401	1.526	-0.624	1.178	-0.733	-0.934	1.979	0.158	1.109	1.516	1.119
1	-0.539	1.319	-0.523	-0.906	-0.348	1.198	-0.625	1.038	-0.663	-0.934	1.257	0.245	1.214	0.144	-0.051
1	-0.662	0.765	-0.922	-0.906	-0.419	0.948	-0.623	1.075	-1.109	-0.934	1.199	0.405	0.973	0.848	-0.350
1	-0.688	2.159	0.907	-0.823	-0.401	-0.432	-0.613	-0.355	1.030	-0.934	0.332	-0.096	2.887	0.726	0.766
1	-0.705	0.622	-1.240	-0.795	-0.410	1.634	-0.615	1.743	-0.584	-0.934	2.181	0.499	0.807	2.158	0.711
2	0.075	-0.489	0.650	0.814	-0.445	0.715	-0.474	0.409	0.218	0.187	-0.506	-0.280	-0.714	-0.991	-1.221
2	-0.022	-0.906	-0.903	3.441	4.352	-0.238	1.479	0.240	1.148	2.592	-0.881	0.333	-0.684	0.525	0.929
2	-0.399	-0.763	-0.687	-0.564	-0.419	-0.701	-0.613	-0.772	-0.762	-0.551	-0.737	-0.719	-0.669	-1.459	-0.459
2	-0.574	-0.466	0.465	1.045	-0.348	0.789	-0.578	0.800	0.426	0.241	-0.534	0.091	-0.699	-0.963	-0.922
2	-0.635	-0.590	-0.247	0.037	-0.189	0.527	-0.598	-0.191	0.099	-0.251	-0.650	-0.546	-0.519	-1.370	-0.786
2	-0.565	-0.814	0.067	0.435	1.516	-1.547	0.849	-0.636	-1.831	0.733	-0.968	0.108	-0.308	0.739	0.439
2	-0.583	-0.763	-0.497	-0.425	-0.348	-0.413	0.955	-0.167	-0.683	-0.415	-0.708	-0.418	-0.654	-0.578	-0.759
2	-0.390	-0.685	-0.165	0.610	0.589	-0.124	0.146	-0.045	-0.198	0.378	-0.708	-0.158	-0.609	-0.586	-0.405

In addition, the table shows the autoscaled brandy data.

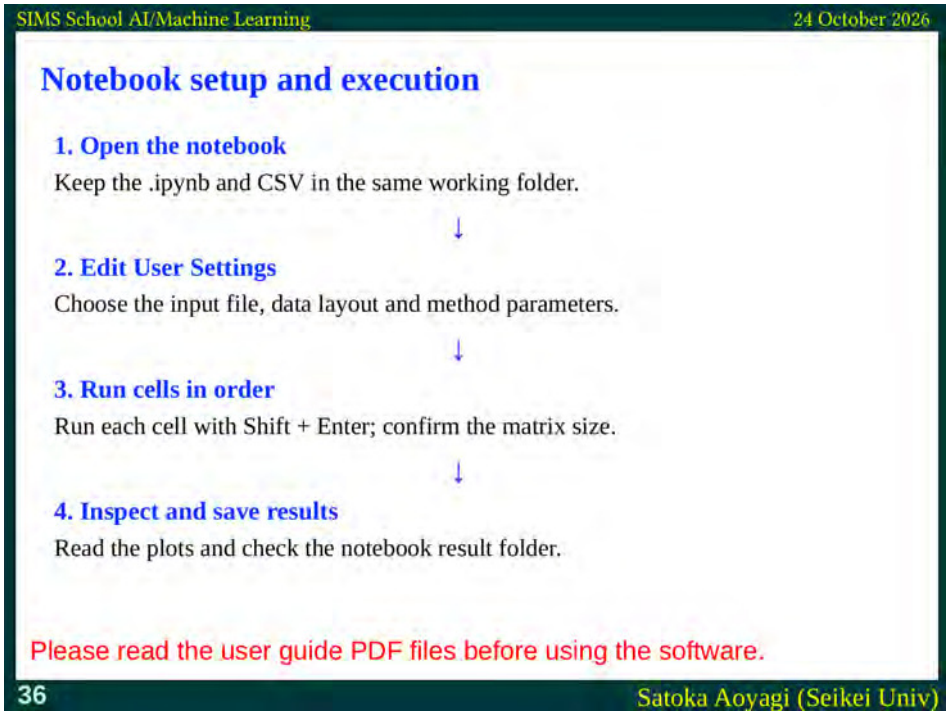
Each variable is scaled by subtracting its mean and dividing by its standard deviation, so variables measured on different numerical scales can be compared. This processing changes the relative influence of the variables on the extracted components.



I've shared the software suite for this tutorial.

The practical suite contains PCA, MCR-ALS and sparse-autoencoder notebooks for unsupervised learning analysis, and Random Forest notebooks for supervised analysis. The RF importance notebooks predict a selected numerical variable or mass image from the other variables.

The additional RF_specBrandy and RF_specPolymer notebooks perform classification. There are separate editions for sample-by-variable data and pixel-by-mass data, with example CSV files and PDF user guides.



To run a supplied notebook, keep it and its CSV data in the same working folder and open the notebook in Jupyter.

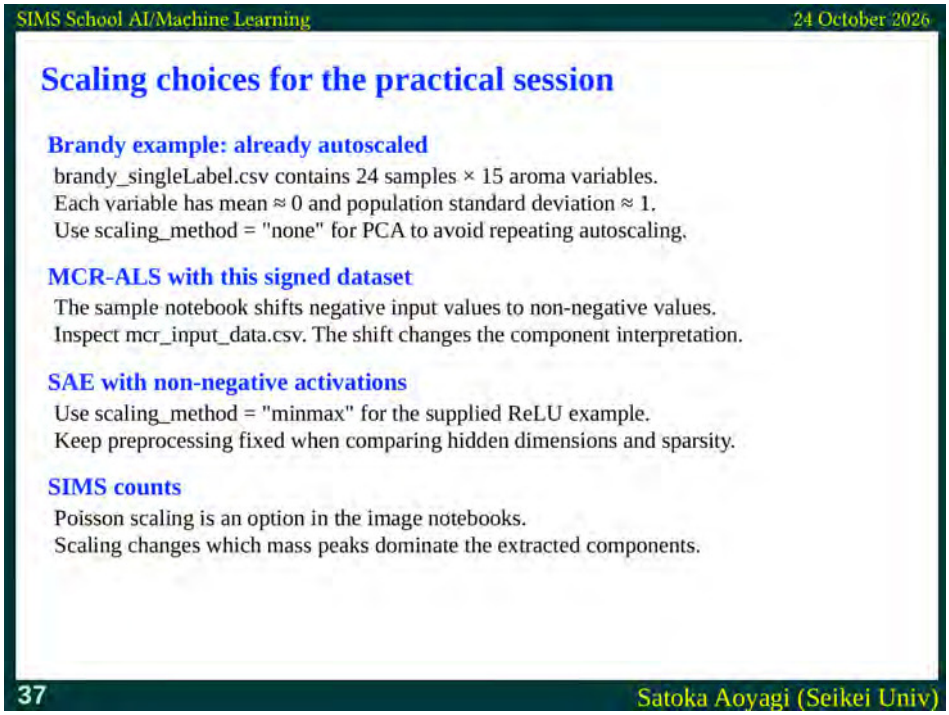
Edit User Settings, then run the cells in order with Shift and Enter.

Check the printed matrix size before interpreting the plots, and confirm where the result files are saved.

SAE requires TensorFlow and Keras, while the image MCR notebook also requires pyMCR.

Sources: supplied SoftwareSuite notebooks and user guides.

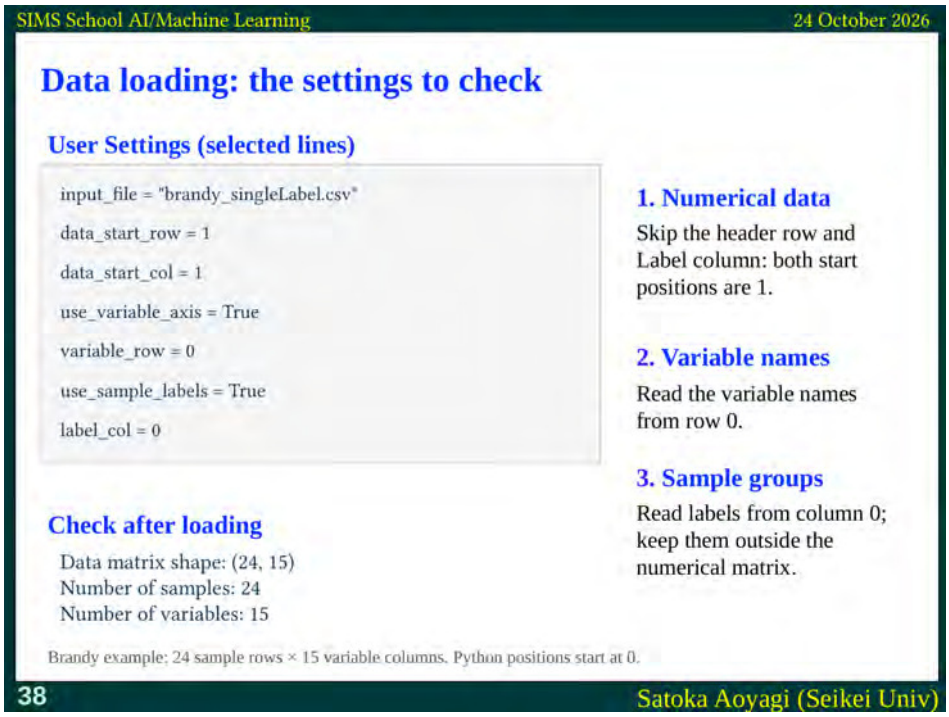
The details are described in the user guide for each program, so please read the relevant guide before using it."



The supplied Brandy CSV is already autoscaled, so PCA uses no additional scaling. MCR-ALS requires a non-negative input. The SAE ReLU example uses min-max scaling. For SIMS count data, the image notebooks also offer Poisson scaling. Keep the preprocessing recorded when comparing methods or parameter settings.

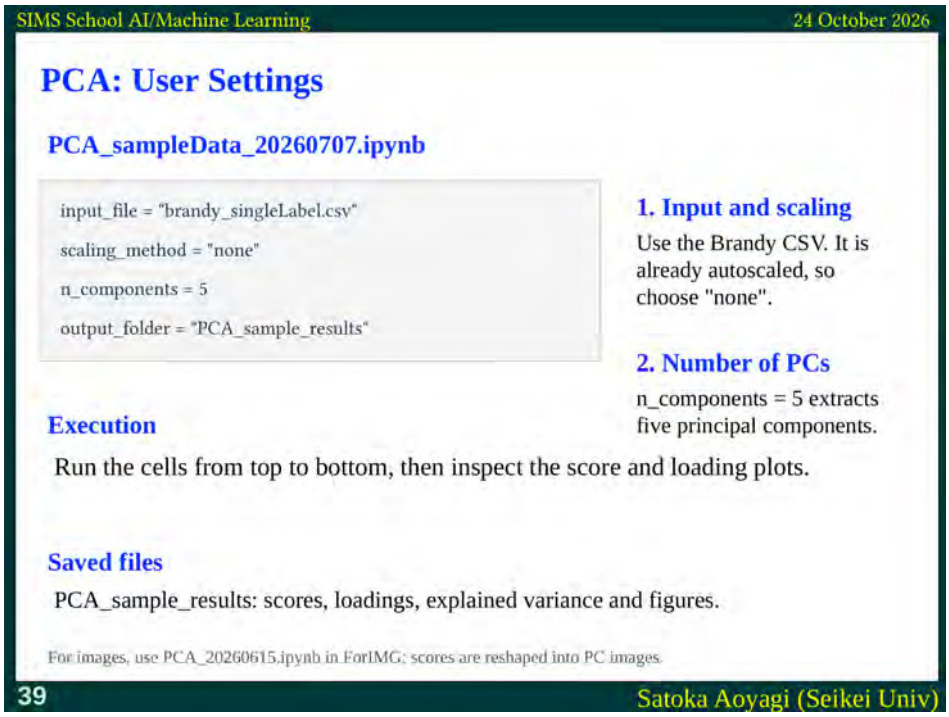
Sources: supplied SoftwareSuite notebooks, example CSV files and 2026 user guides.

The details are described in the user guide for each program, so please read the relevant guide before using it."



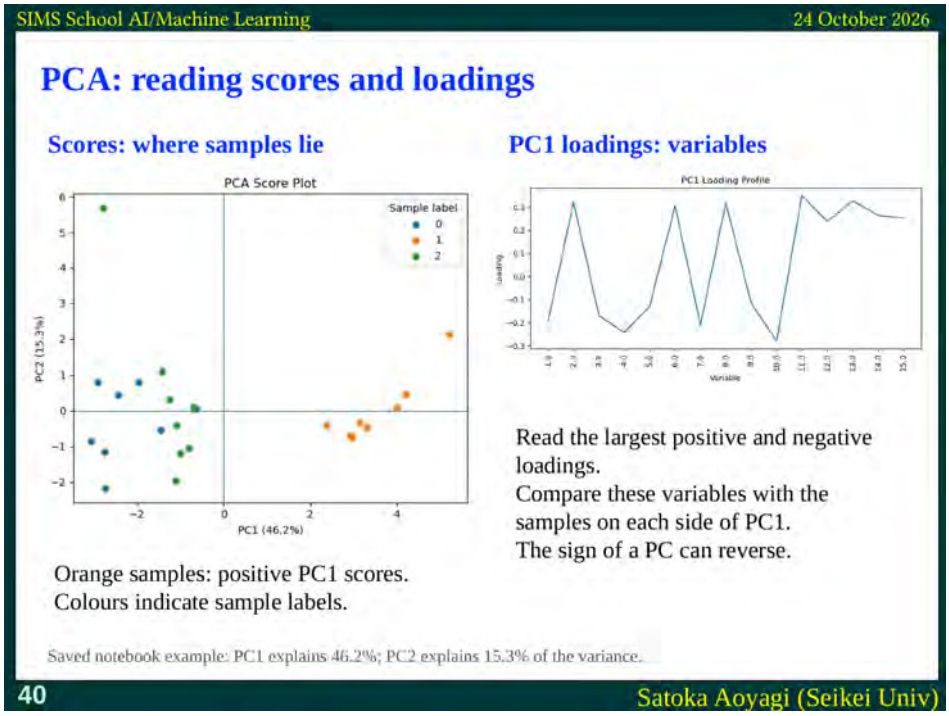
These are the data-loading settings for the Brandy example. Python positions start at zero, so row one and column one skip the header and Label column. Read the variable names from row zero and the sample labels from column zero. After loading, confirm a numerical matrix of 24 rows and 15 columns before continuing.

Source: PCA_sampleData_20260707.ipynb, User Settings and data-loading output.



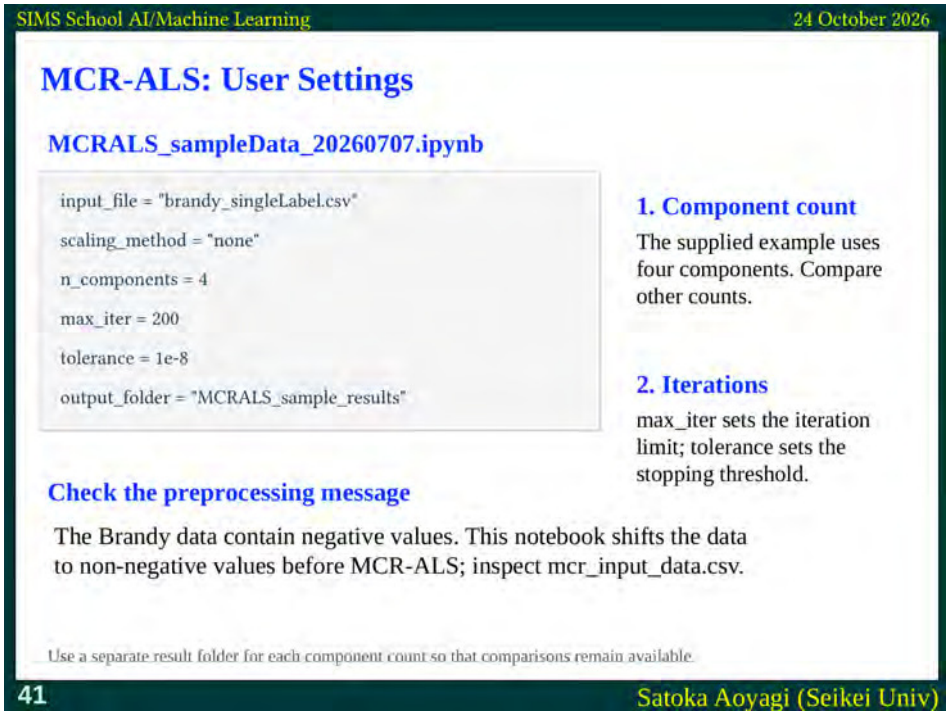
For PCA, select the Brandy input file and retain scaling_method as none because the file is already autoscaled. n_components equal to five extracts five principal components. Run the notebook in order and inspect the explained variance, scores and loadings. The results are saved in PCA_sample_results. The separate image notebook reshapes the scores into PC images.

Source: PCA_sampleData_20260707.ipynb, cell 1.



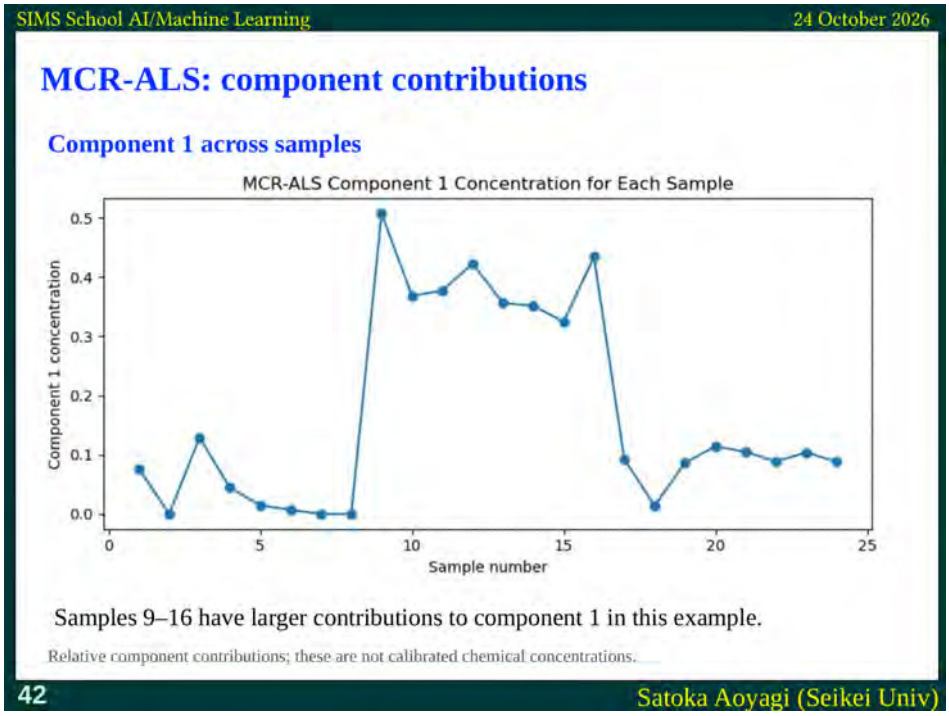
Read the score and loading plots together. The orange samples lie on the positive PC1 side in this saved result. The largest positive and negative PC1 loadings identify variables associated with the two directions. PC1 explains 46.2 percent of the variance and PC2 explains 15.3 percent. A principal-component sign can reverse, so interpret the relationship between scores and loadings rather than the sign alone.

Source: PCA_sampleData_20260707.ipynb, saved outputs in cells 3 and 5.



The MCR-ALS sample settings specify four components and a limit of 200 iterations. The tolerance controls the stopping criterion. Because the Brandy data contain negative values, inspect the preprocessing message and mcr_input_data.csv to confirm the shift applied before fitting. Compare component counts using separate result folders and evaluate the component profiles alongside reconstruction error.

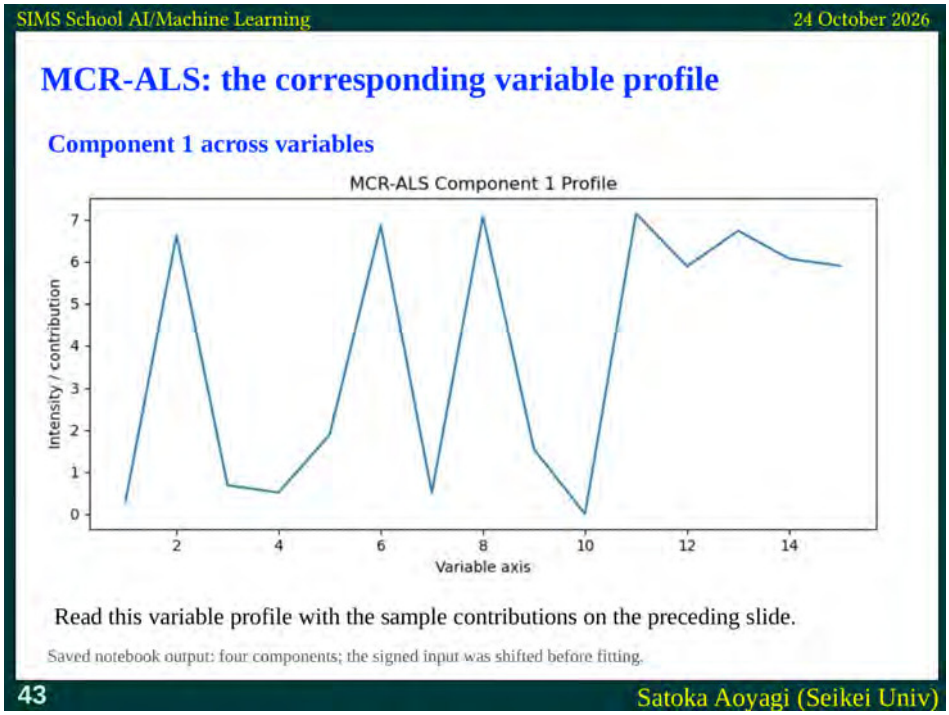
Source: MCRALS_sampleData_20260707.ipynb, cells 1, 3 and 4.



This plot shows the contribution of component one across the samples. Samples nine through sixteen, the cognac samples, have relatively higher values in this example. These values represent relative component contributions rather than calibrated chemical concentrations.

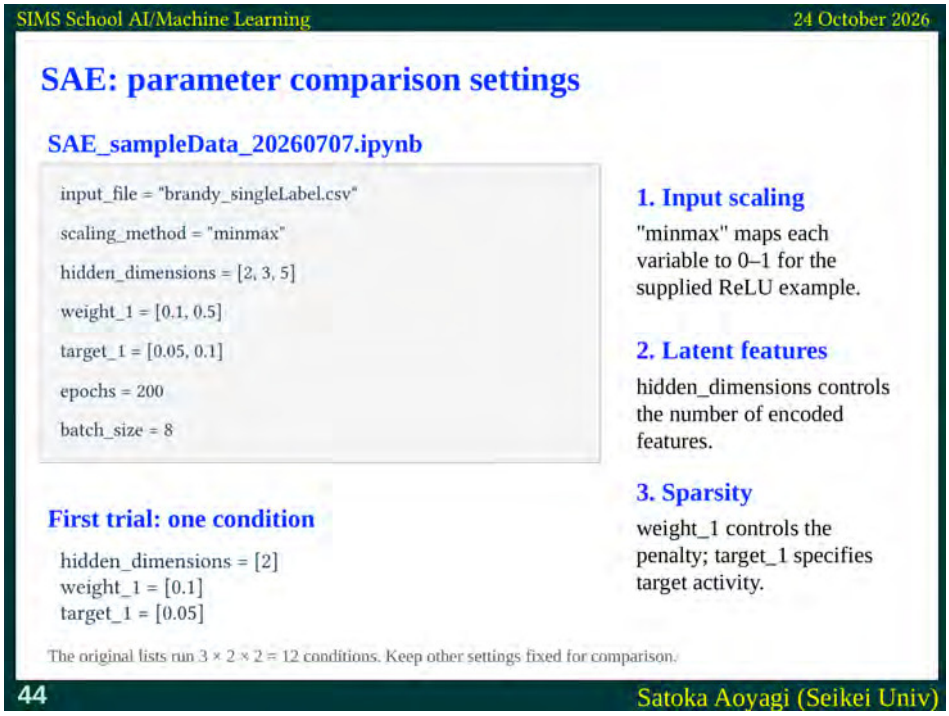
The next slide shows the variable profile for the same component.

Source: MCRALS_sampleData_20260707.ipynb, component 1 saved output in cell 6.



This is the variable profile of component one. The larger values indicate variables contributing more strongly to this component. Compare this profile with the sample contributions on the preceding slide to identify which measured compounds support the sample pattern. The non-negative shift applied before fitting must be considered when interpreting these values.

Source: MCRALS_sampleData_20260707.ipynb, component 1 saved output in cell 7.

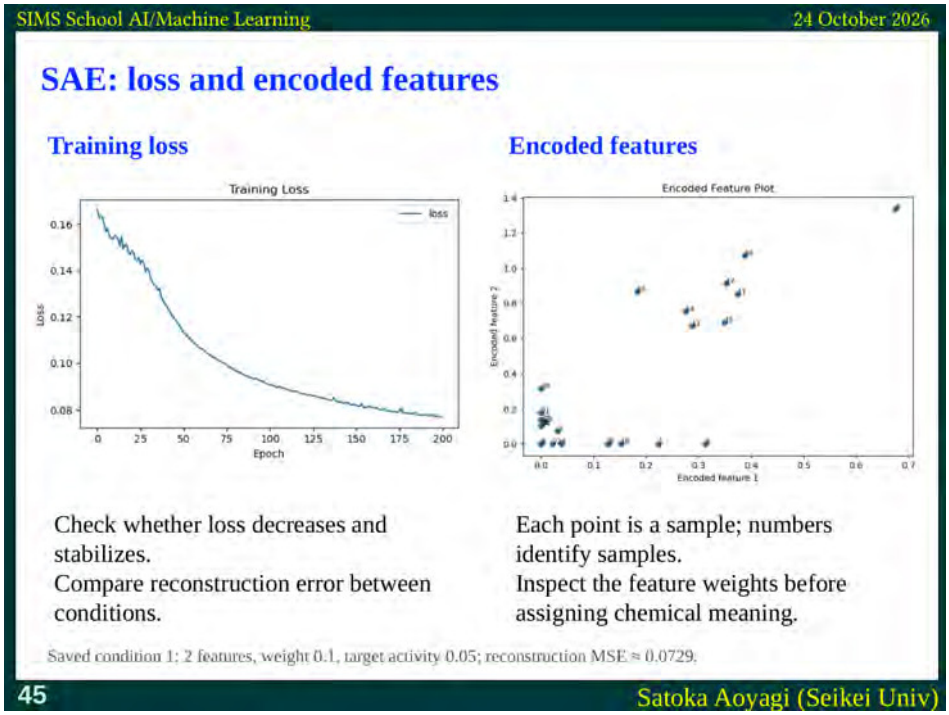


The SAE settings use min-max scaling and compare hidden-layer dimensions of two, three, and five. The sparsity weight controls the penalty strength, while the target activity specifies the desired average activation.

The program tests all twelve combinations in the supplied parameter lists. For a quick initial run, use the single-value lists shown below the main settings.

Compare the results and select the most promising settings. If needed, refine them by testing nearby parameter values.

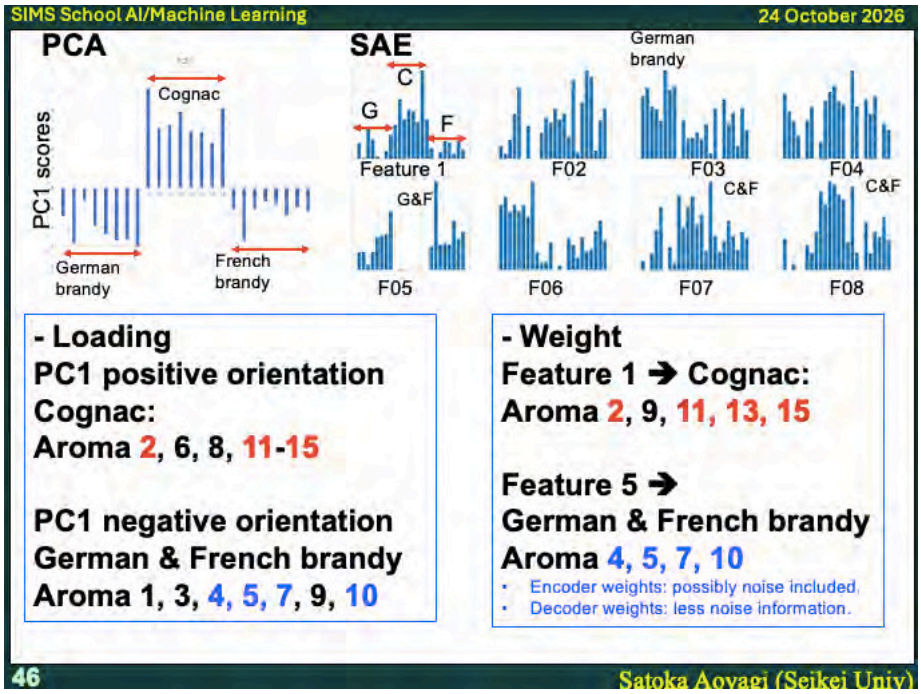
Source: SAE_sampleData_20260707.ipynb, cell 1.



This saved example uses two encoded features, a sparsity weight of 0.1 and a target activity of 0.05.

The training-loss curve decreases and approaches a stable value. The encoded-feature plot positions the samples in the learned feature space, with sample numbers beside the points. Compare reconstruction error across conditions and inspect feature weights before assigning chemical meaning to a separation.

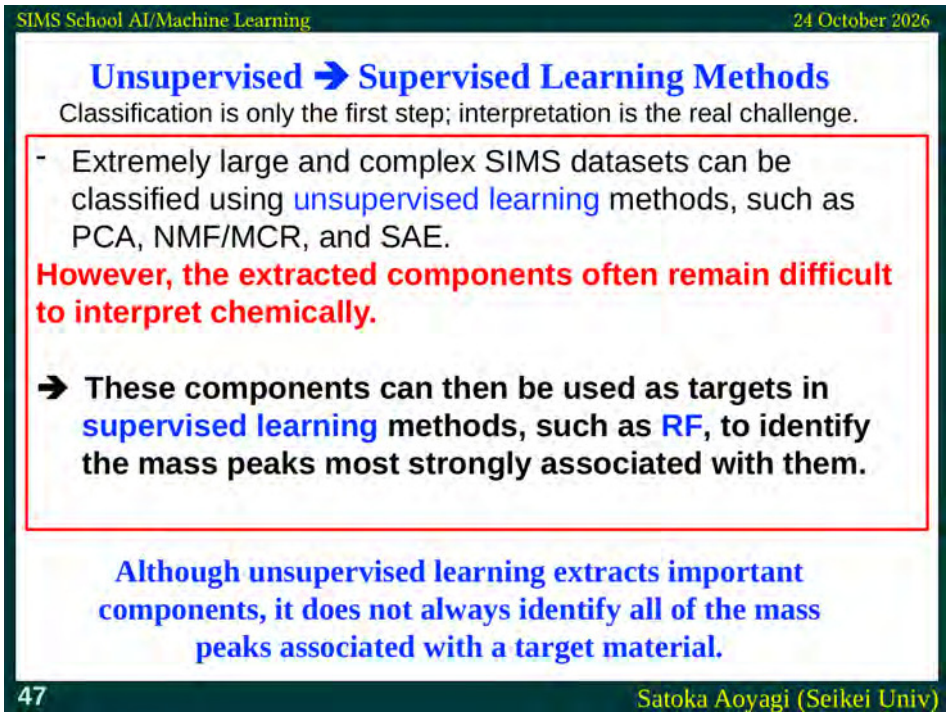
Source: SAE_sampleData_20260707.ipynb, cell 5, saved condition 1: hid=2, weight=0.



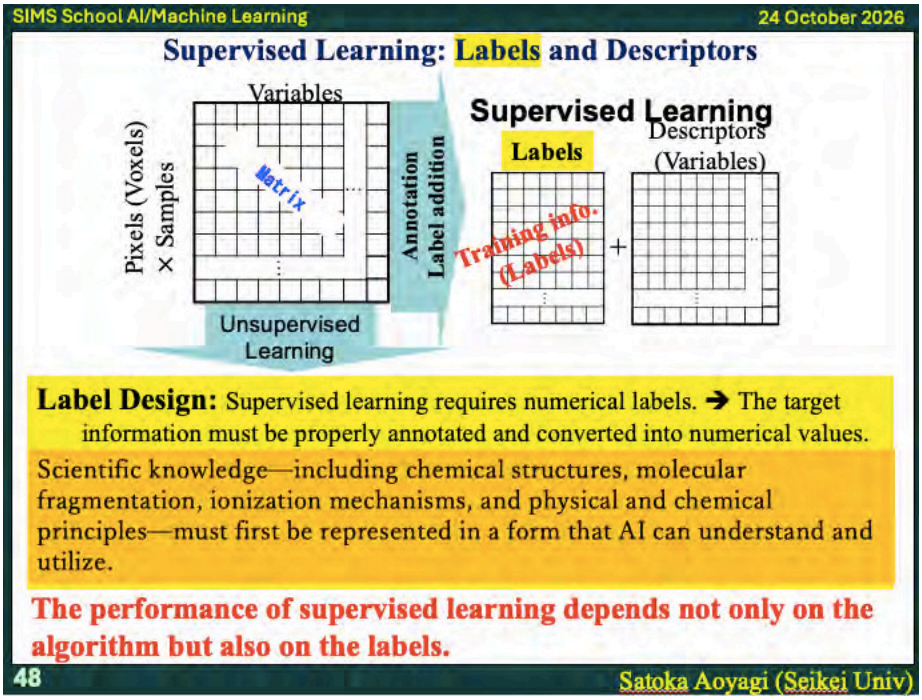
This is a separate Brandy analysis example using an eight-feature SAE result.

PCA separates Cognac from the German and French brandies along PC1, and the loadings identify the aroma compounds supporting this direction.

SAE feature one relates to Cognac, while feature five relates to the German and French groups. Compare the listed weights with the PCA loadings. The parameter settings differ from the two-feature practical example shown earlier.

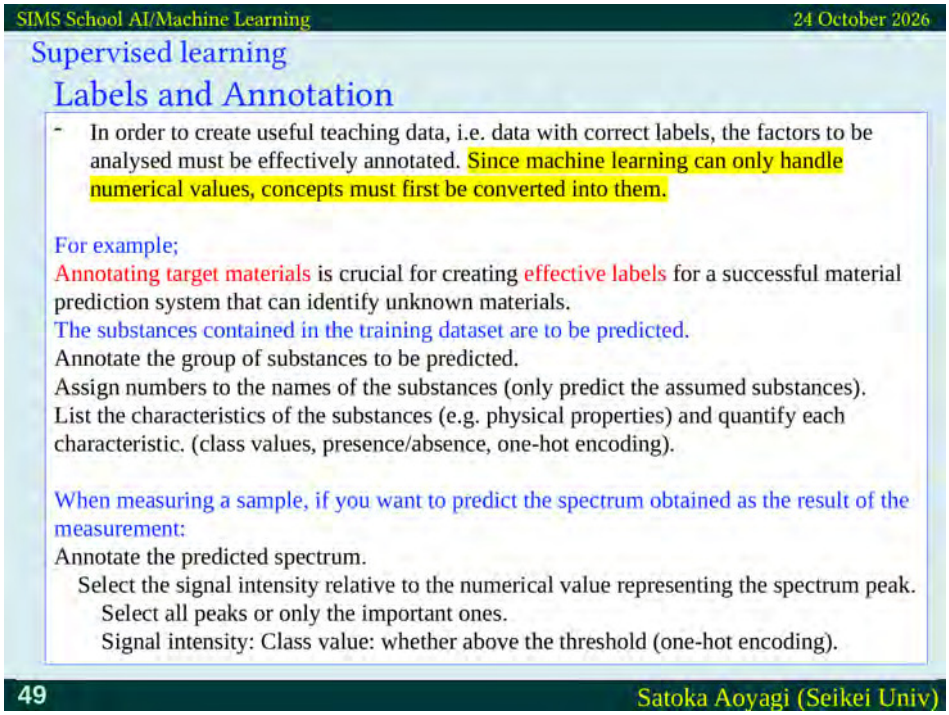


Unsupervised learning gives us component patterns or features, but their chemical identity can remain uncertain. We can define an interesting peak, spatial pattern or extracted component as a supervised-learning target and investigate the measured variables related to it. The following section first explains target labels and the RF mechanism, then demonstrates the practical notebooks.

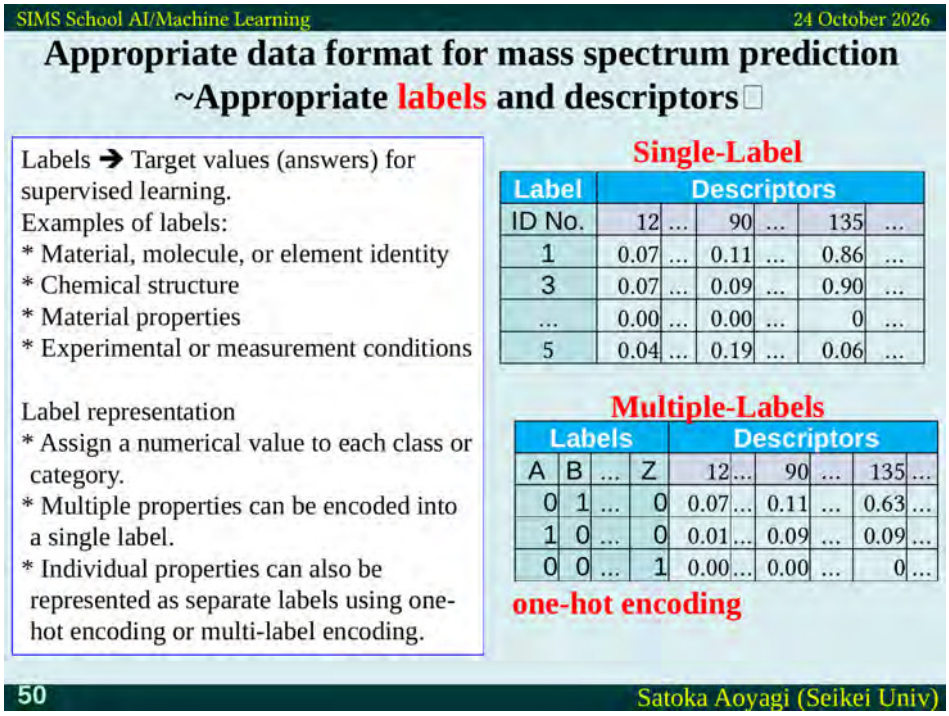


Supervised learning adds target information to the measured descriptor matrix.

For example, for material prediction, labels can represent material identities or properties. For spectrum-related prediction, they can represent peak intensities or defined classes. This connects scientific knowledge with the specific question that the model learns to answer.

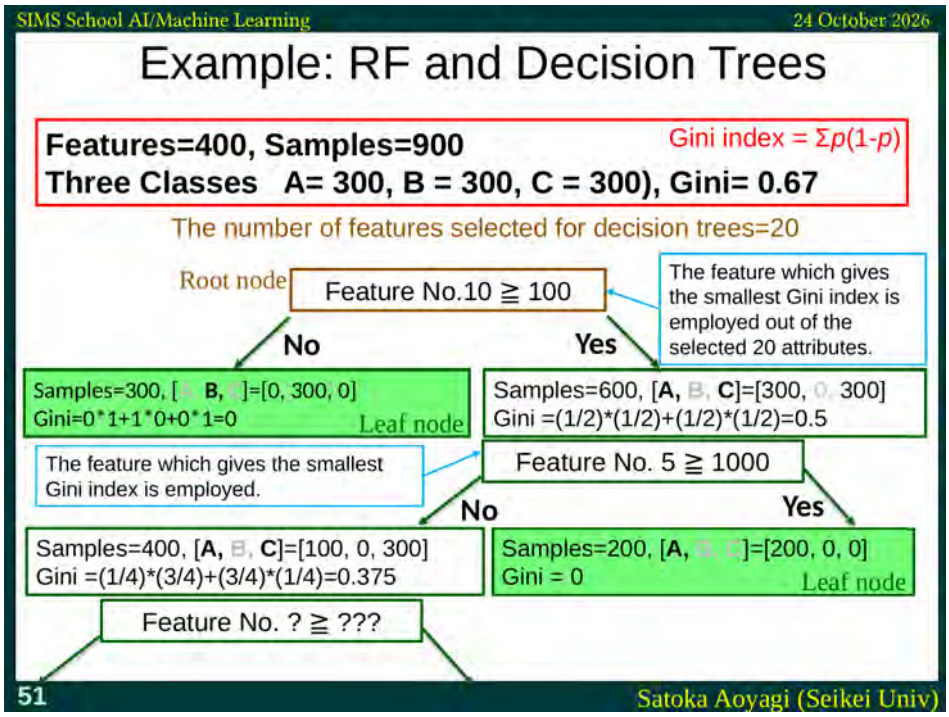


Annotation specifies what the target labels mean. Assigning an ID to each known entity supports classification among those entities. Encoding shared properties or structural elements can support a more general prediction task. For target spectra, we can use measured intensities or threshold-based classes. These representations define the scope of the model before training begins.

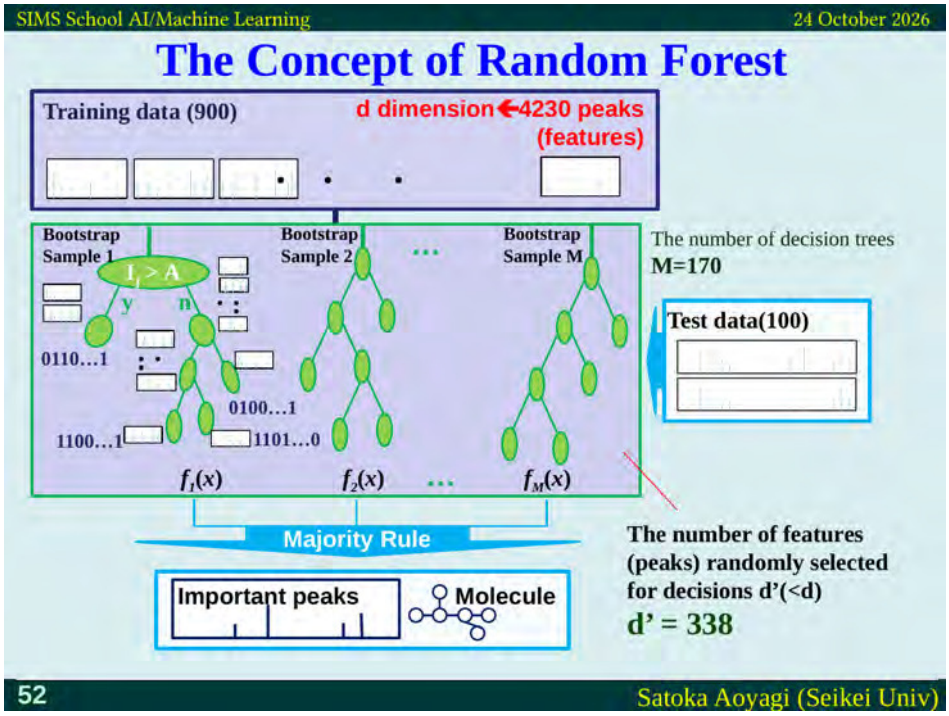


The two tables compare a single target-label column with multiple target-label columns.

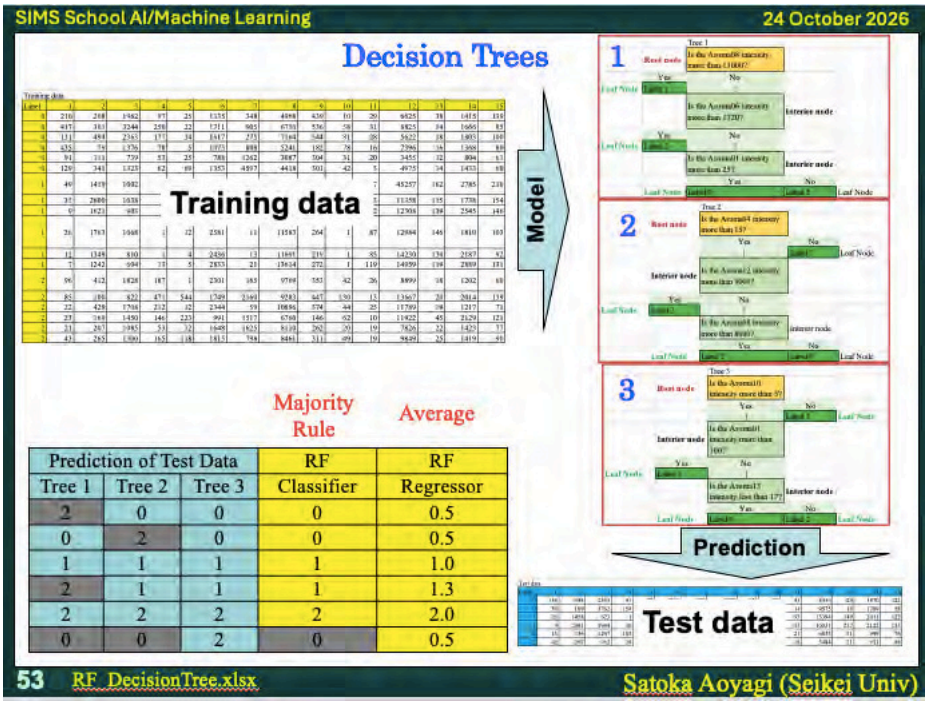
For example, in material prediction, a material ID represents a fixed category, while separate presence-or-absence labels can represent individual properties or structural elements. The numbers used as category IDs are categorical codes. Shared structural labels can help us investigate materials beyond a fixed list, provided the training data cover the relevant structures.



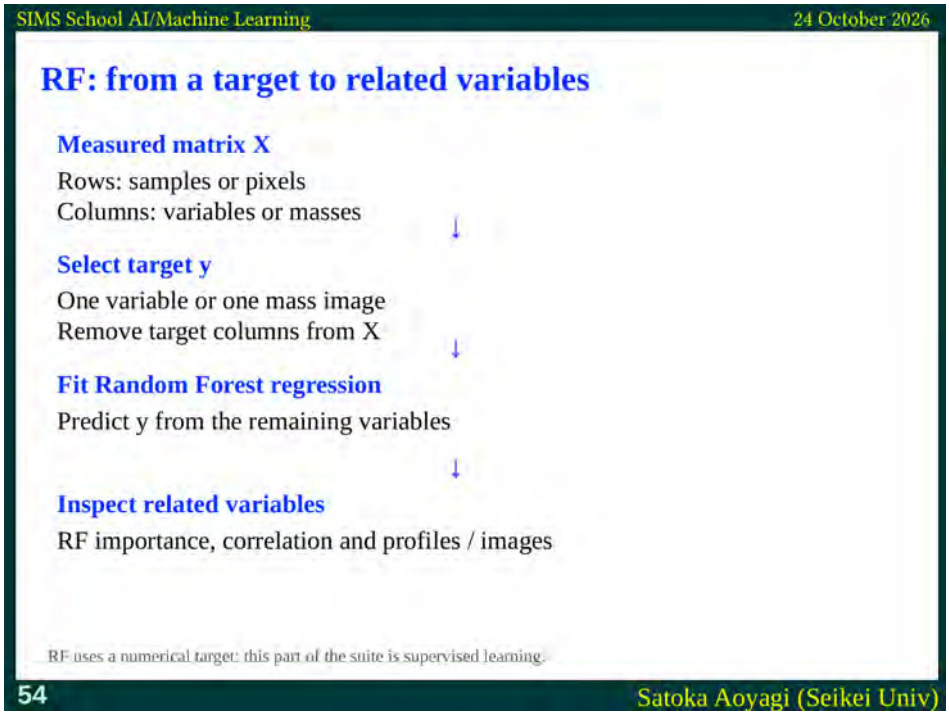
A decision tree separates samples by asking questions about feature values. In this example, the questions compare peak-related variables with thresholds. The Gini index describes how mixed the classes are at a node; it is zero when all samples belong to one class. A candidate split is evaluated by how much it reduces the weighted impurity of the resulting nodes. Repeating this process produces branches and eventually leaf nodes with class predictions.



This example illustrates material prediction using a random forest. A random forest trains many trees on bootstrap samples and considers a random subset of variables for the candidate splits. The ensemble combines the resulting predictions. The numbers of peaks, selected features, and trees shown here belong to this particular example and are not universal settings.

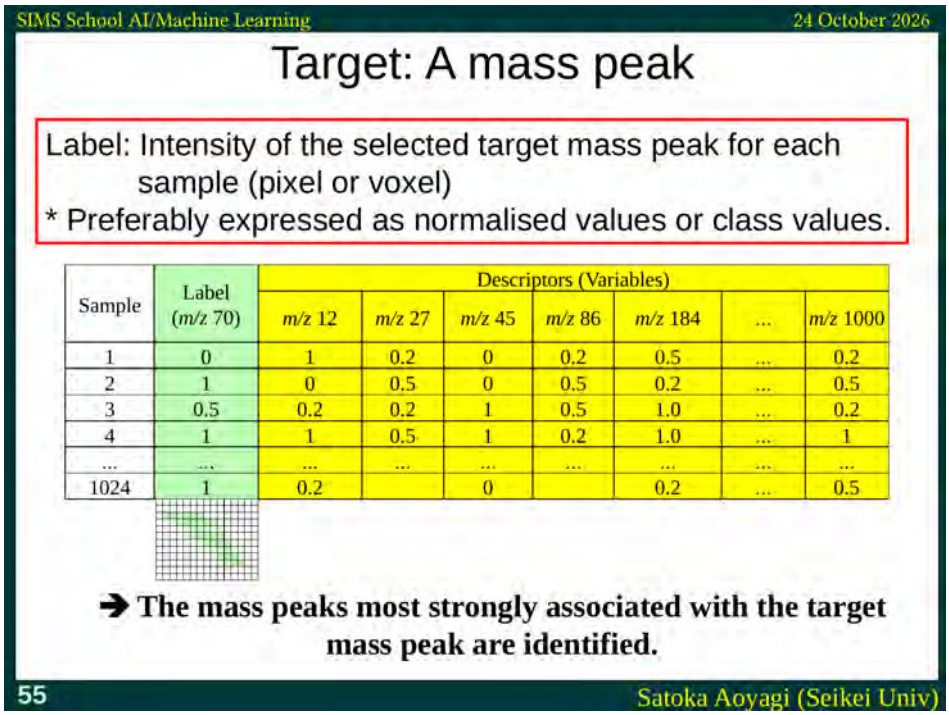


This spreadsheet shows how the forest combines predictions from individual trees. Classification uses voting among class predictions, while regression averages numerical predictions. Keep training and test data separate when evaluating the resulting model.

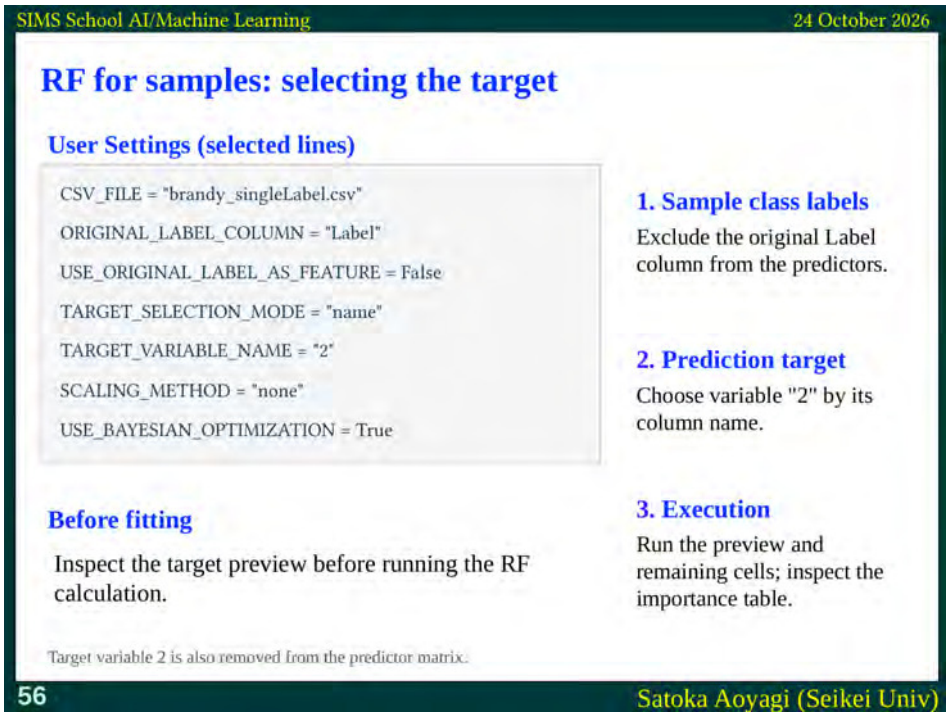


Random forests can be used for material prediction through classification and for quantitative analysis through regression. They can also identify variables that are important for predicting a selected target. This flowchart explains how to use the random forest importance programs for this purpose.

Source: RF_importanceSPEC_20260707.ipynb and RF_importanceIMG_20260707.



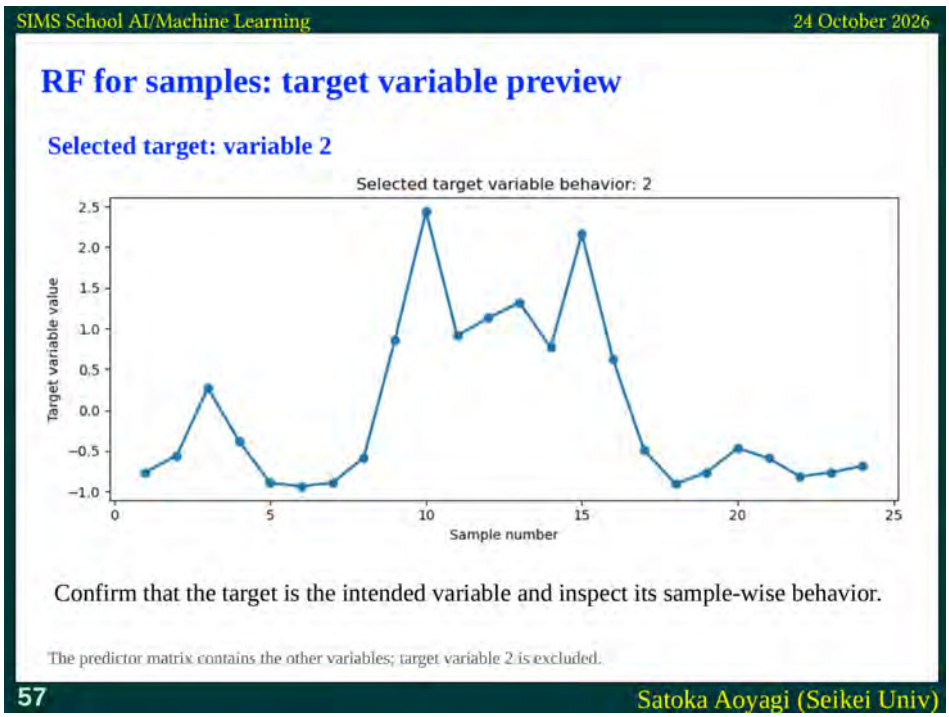
In the mass-peak example, the intensity of a selected peak is the target value for each sample or pixel. The remaining mass peaks are used as predictors. We exclude the target peak from the predictors so that the model examines its relationships with other variables. The target remains numerical for regression. Alternatively, we can define classes using an intensity threshold and perform classification.



I'd like to show you how to use RF importance programs using the Brandy data example.

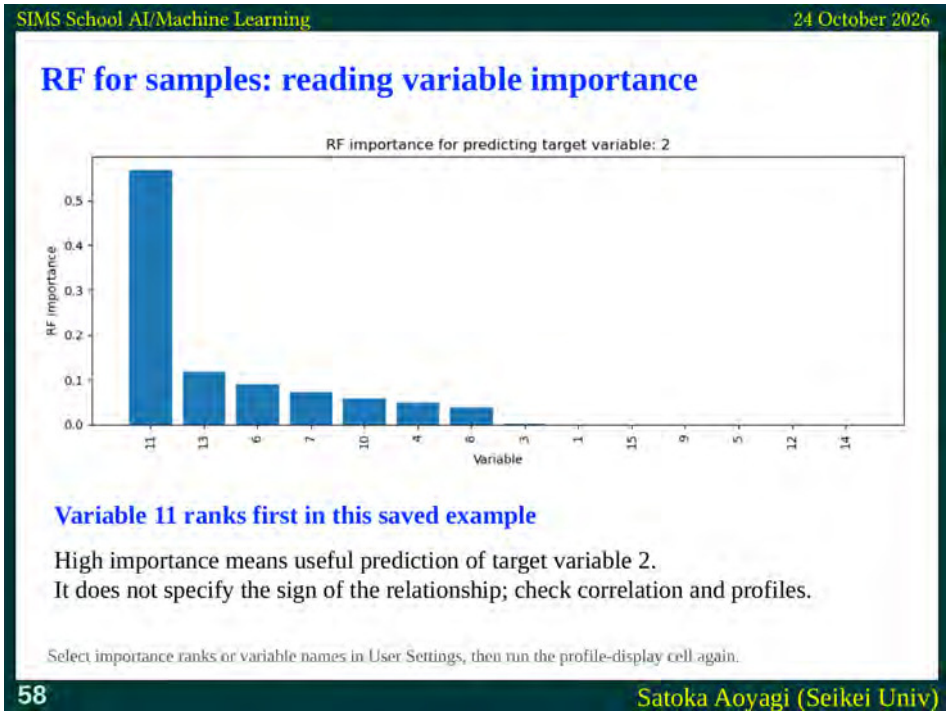
The sample RF notebook uses the Brandy CSV and selects target variable two by its column name. Keep USE_ORIGINAL_LABEL_AS_FEATURE false to exclude the original group labels. The target variable is also excluded from the predictors. Check the preview, then run the remaining cells for parameter selection, fitting and importance analysis.

Source: RF_importanceSPEC_20260707.ipynb, User Settings and saved target preview.



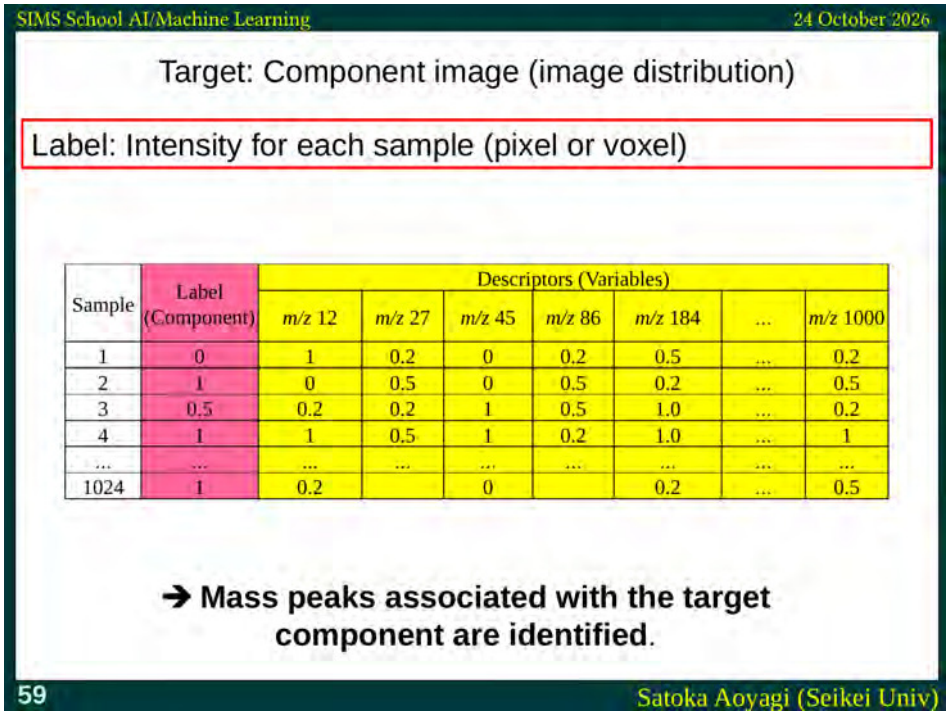
For the brandy example, this preview shows the selected target, variable two, across the samples. Confirm that the intended column has been selected and examine its sample pattern before fitting the model. The random forest will predict this target using the remaining measured variables.

Source: RF_importanceSPEC_20260707.ipynb, saved target variable preview in cell 2.



In this saved example, variable eleven has the highest RF importance for predicting variable two. Importance describes usefulness for prediction and doesn't indicate whether the relationship is positive or negative. Inspect the correlations and the selected variable profiles as well. To examine other variables, change the ranks or variable names in the user settings and rerun the cell that displays the profiles.

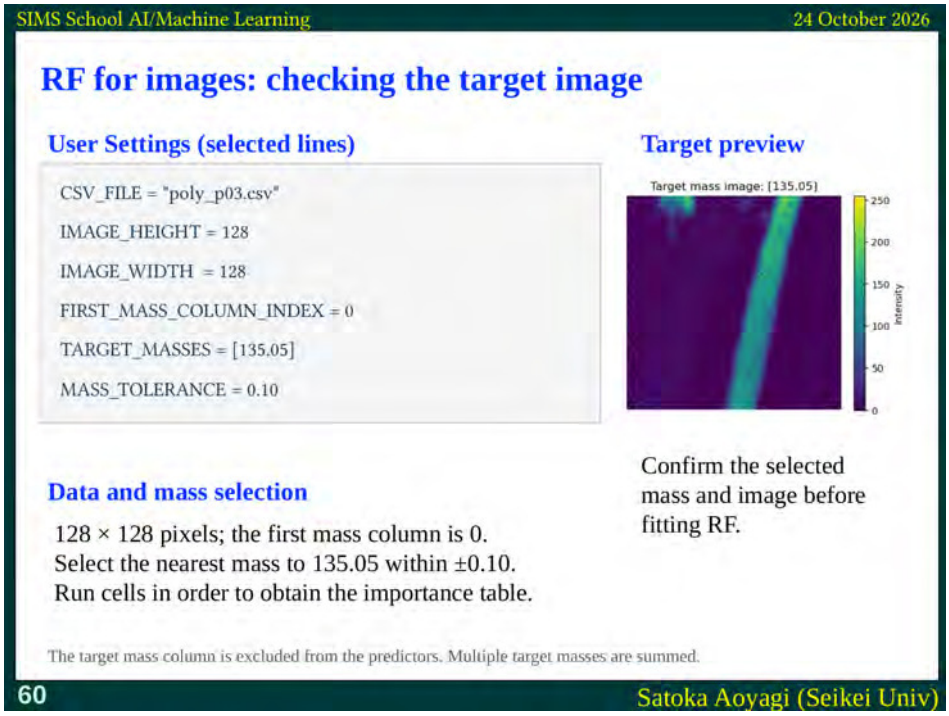
Source: RF_importanceSPEC_20260707.ipynb, cell 7 saved RF importance plot.



RF importance can help us find variables associated with a particular region of the sample.

An extracted component image can also be used as a numerical target. Its value at each pixel becomes the target value, y, while the original mass-peak intensities form the predictor matrix, X.

This is an extension of the workflow. The supplied imaging RF notebook uses selected mass columns or their sum as the target. Using a PCA, MCR, or SAE component image therefore requires adapting the target input. The resulting importance indicates which mass peaks are useful for predicting the component's spatial distribution.

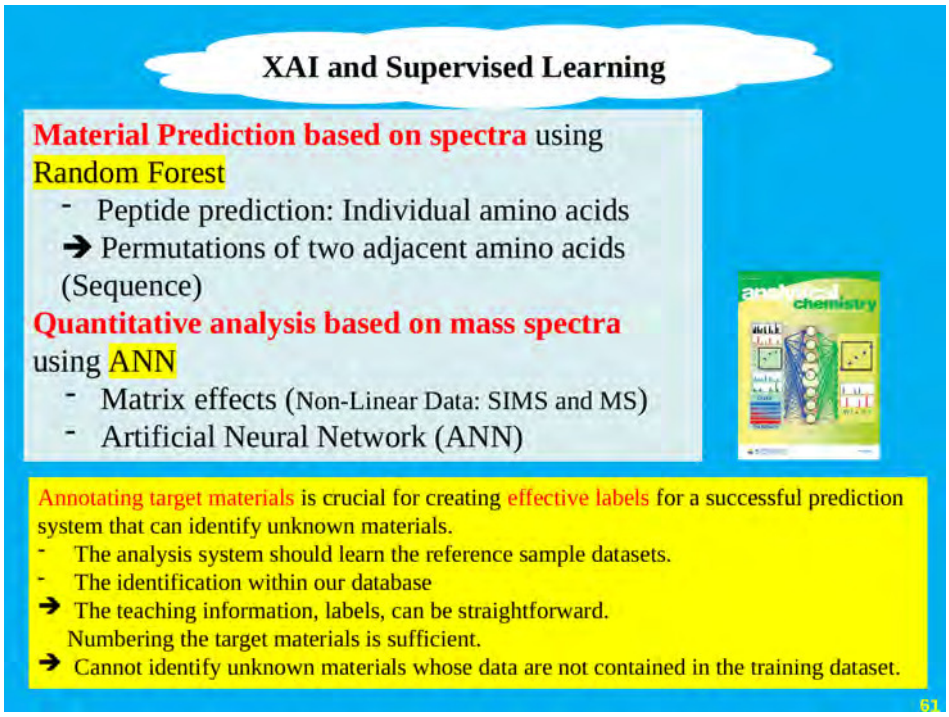


You can use either the polymer dataset with 999 mass peaks or the one with 570 mass peaks. Select a component image obtained from PCA, MCR, or SAE and, after adapting the notebook's target input, use its pixel values as the target for random forest regression.

The original mass-peak intensities serve as predictors. RF importance then identifies the mass peaks that are most useful for predicting the target image's spatial pattern. This is one way to use random forests to investigate variables associated with an extracted component.

In this example, the mass peak at m/z 135 was identified as an important predictor of the target region's spatial pattern.

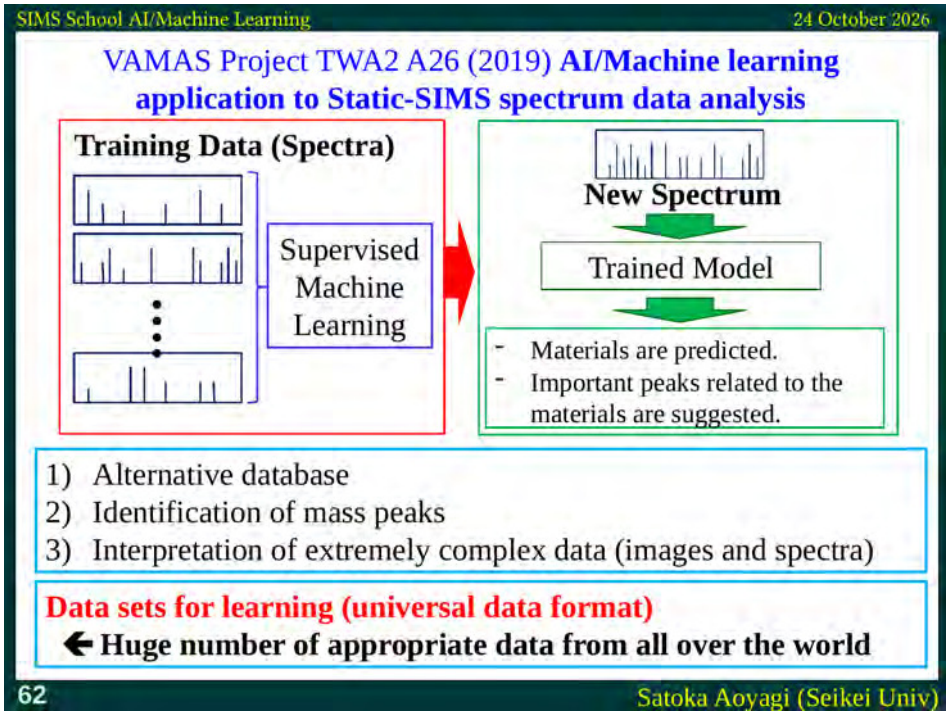
Source: RF_importanceIMG_20260707.ipynb, User Settings and saved target preview in cell 2.



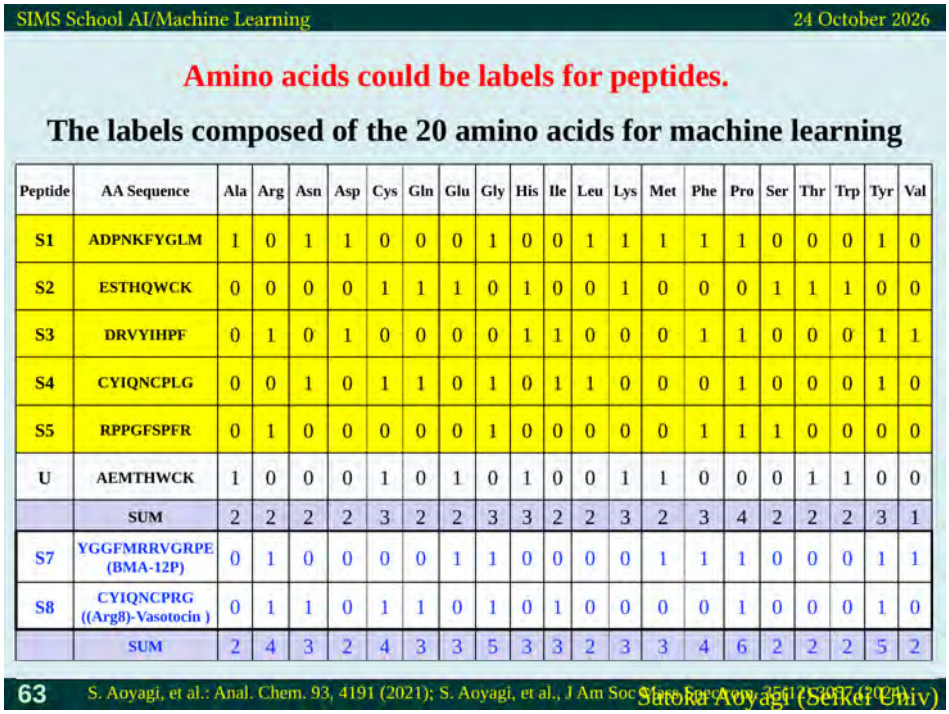
We now move to two applications: material prediction from spectra and quantitative analysis of mixtures.

For material prediction, labels represent chemical structures, starting with amino acids and adjacent amino acid pairs. Assigning IDs to known materials supports classification within a fixed database, while structural labels describe information shared by different molecules.

For quantitative analysis, labels represent composition.



We started the VAMAS project for material prediction in 2019----

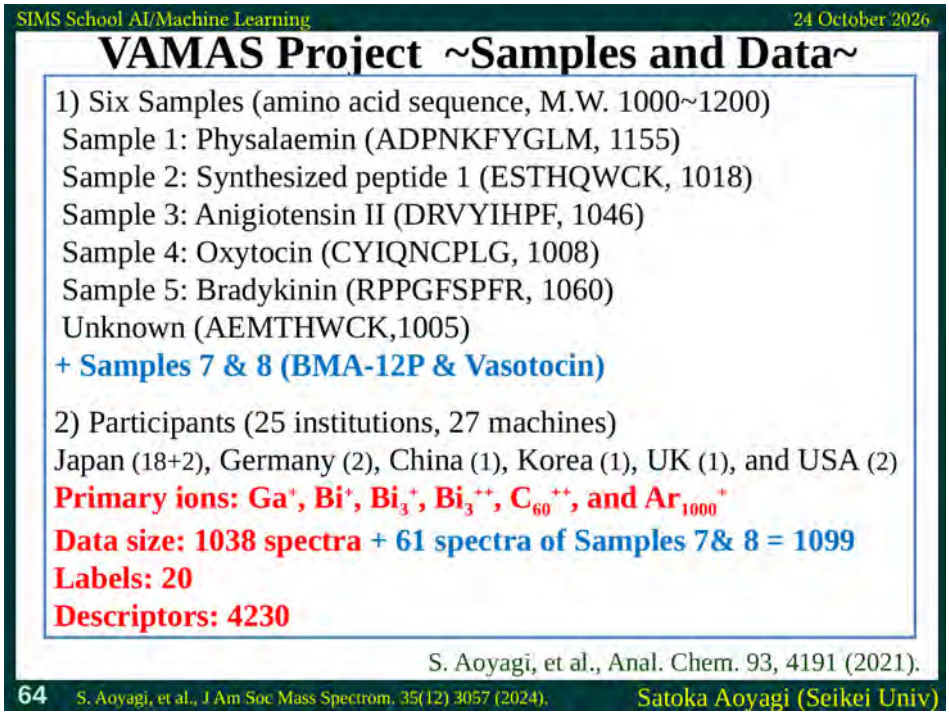


Peptide	AA Sequence	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
S1	ADPNKFYGLM	1	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1	0
S2	ESTHQWCK	0	0	0	0	1	1	1	0	1	0	0	1	0	0	0	1	1	1	0	0
S3	DRVYIHPF	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1
S4	CYIQNCPLG	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1	0	0	0	1	0
S5	RPPGFSPFR	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0
U	AEMTHWCK	1	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	0
	SUM	2	2	2	2	3	2	2	3	3	2	2	3	2	3	4	2	2	2	3	1
S7	YGGFMRRVGRPE (BMA-12P)	0	1	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1
S8	CYIQNCPRG ((Arg8)-Vasotocin)	0	1	1	0	1	1	0	1	0	1	0	0	0	0	1	0	0	0	1	0
	SUM	2	4	3	2	4	3	3	5	3	3	2	3	3	4	6	2	2	2	5	2

Initially, we chose peptides as model samples because they contain combinations of the twenty amino acids. We used twenty labels to represent the presence or absence of each amino acid.

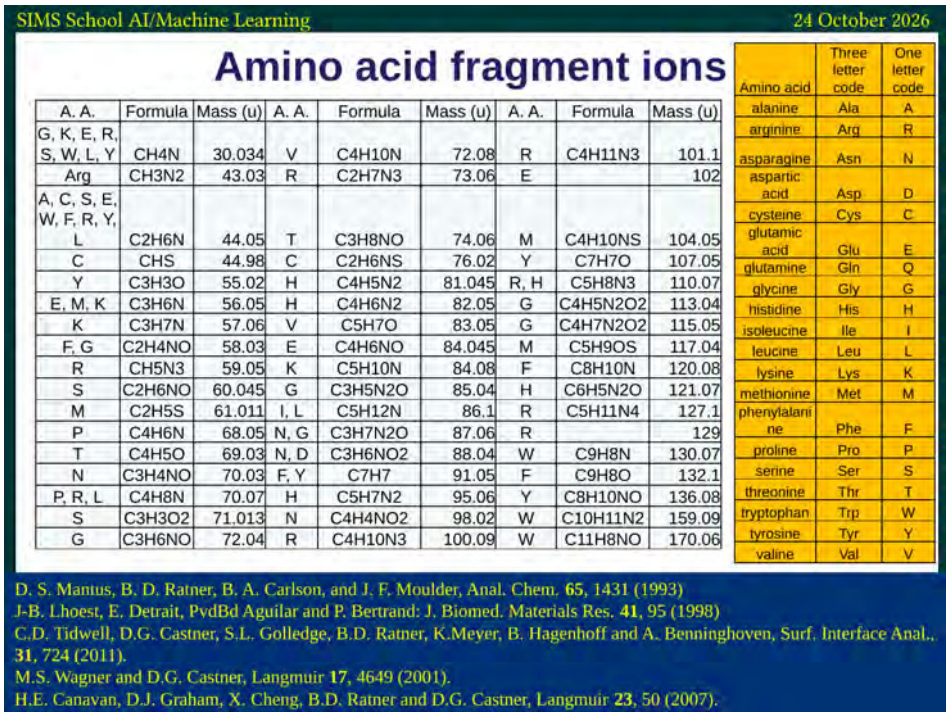
The label table encodes the presence or absence of each of the twenty amino acids in a peptide. One means present, and zero means absent. These shared structural building blocks replace a label that represents only the peptide name.

Adding peptides increases the diversity of the training data, but presence-or-absence labels do not describe residue order.



The VAMAS peptide dataset combines measurements from multiple institutions and primary-ion sources.

The slide lists the peptide samples. The expanded dataset contains 1,099 spectra, with twenty amino-acid labels and 4,230 spectral descriptors. It lets us study prediction across varied measurement conditions.

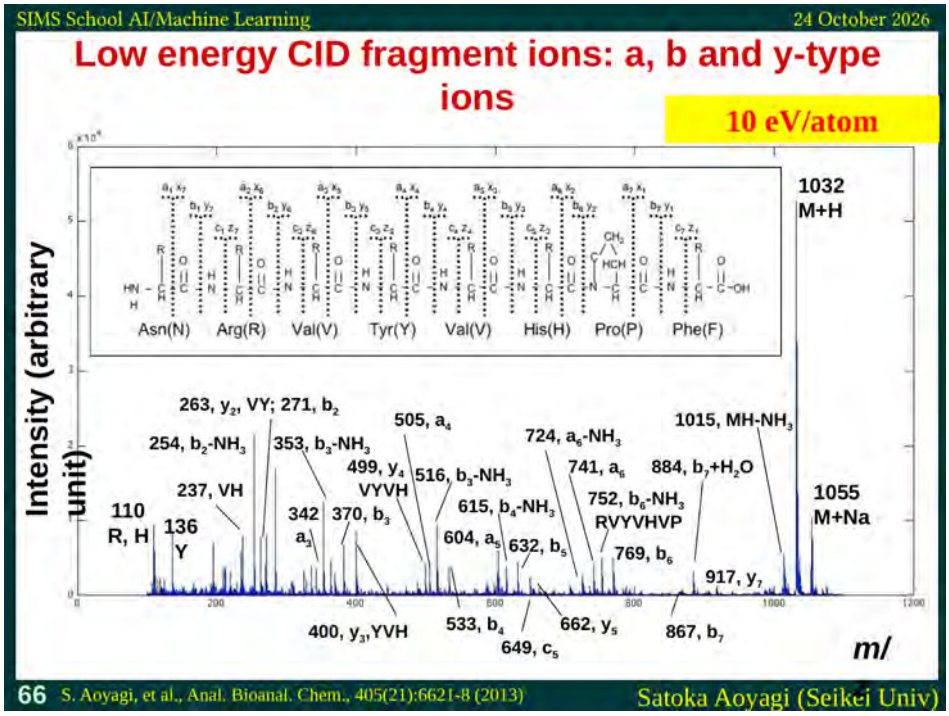


A. A.	Formula	Mass (u)	A. A.	Formula	Mass (u)	A. A.	Formula	Mass (u)
G, K, E, R, S, W, L, Y	CH4N	30.034	V	C4H10N	72.08	R	C4H11N3	101.1
Arg	CH3N2	43.03	R	C2H7N3	73.06	E		102
A, C, S, E, W, F, R, Y, L	C2H6N	44.05	T	C3H8NO	74.06	M	C4H10NS	104.05
C	CHS	44.98	C	C2H6NS	76.02	Y	C7H7O	107.05
Y	C3H3O	55.02	H	C4H5N2	81.045	R, H	C5H8N3	110.07
E, M, K	C3H6N	56.05	H	C4H6N2	82.05	G	C4H5N2O2	113.04
K	C3H7N	57.06	V	C5H7O	83.05	G	C4H7N2O2	115.05
F, G	C2H4NO	58.03	E	C4H6NO	84.045	M	C5H9OS	117.04
R	CH5N3	59.05	K	C5H10N	84.08	F	C8H10N	120.08
S	C2H6NO	60.045	G	C3H5N2O	85.04	H	C6H5N2O	121.07
M	C2H5S	61.011	I, L	C5H12N	86.1	R	C5H11N4	127.1
P	C4H6N	68.05	N, G	C3H7N2O	87.06	R		129
T	C4H5O	69.03	N, D	C3H6NO2	88.04	W	C9H8N	130.07
N	C3H4NO	70.03	F, Y	C7H7	91.05	F	C9H8O	132.1
P, R, L	C4H8N	70.07	H	C5H7N2	95.06	Y	C8H10NO	136.08
S	C3H3O2	71.013	N	C4H4NO2	98.02	W	C10H11N2	159.09
G	C3H6NO	72.04	R	C4H10N3	100.09	W	C11H8NO	170.06

Amino acid	Three letter code	One letter code
alanine	Ala	A
arginine	Arg	R
asparagine	Asn	N
aspartic acid	Asp	D
cysteine	Cys	C
glutamic acid	Glu	E
glutamine	Gln	Q
glycine	Gly	G
histidine	His	H
isoleucine	Ile	I
leucine	Leu	L
lysine	Lys	K
methionine	Met	M
phenylalanine	Phe	F
proline	Pro	P
serine	Ser	S
threonine	Thr	T
tryptophan	Trp	W
tyrosine	Tyr	Y
valine	Val	V

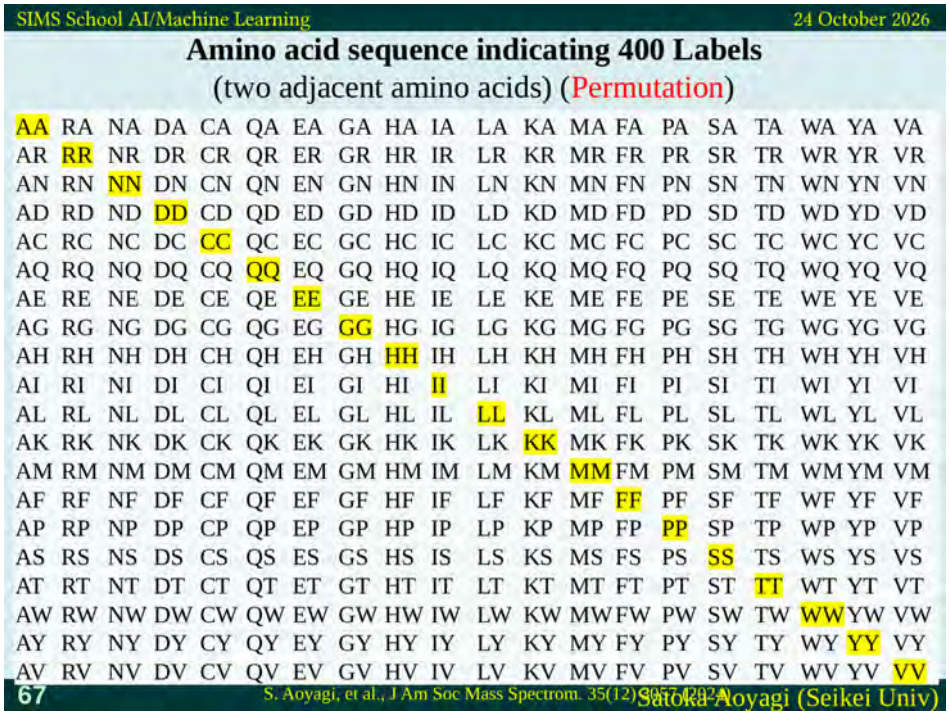
Amino-acid fragment ions produce strong signals in conventional TOF-SIMS, and prediction using these labels worked well. However, presence-or-absence information alone cannot tell us the amino acid sequence. We therefore needed additional information about residue order.

This table lists reported amino-acid-related fragment ions and their literature sources.



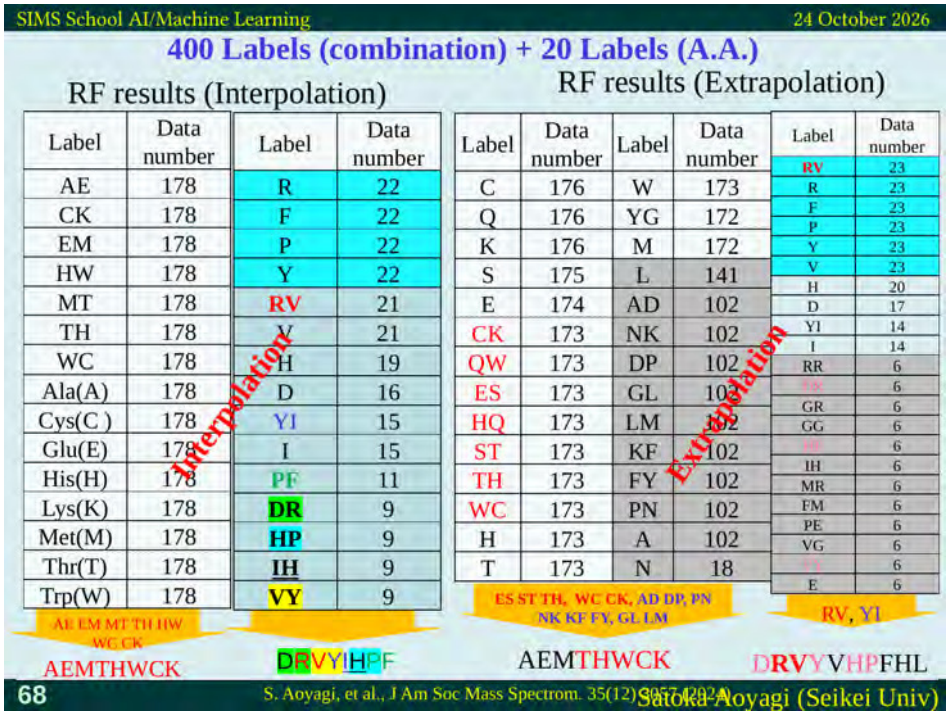
In separate studies, we found many fragment ions containing multiple amino acid residues in SIMS spectra.

This slide shows an example of a peptide spectrum measured using an argon-cluster ion beam. There are several fragment-ion types, including ions containing more than one residue. These larger fragments are important because they can carry information about relationships between neighboring residues, beyond the presence of individual amino acids.



To identify peptides, amino acid sequence information is required. Therefore, we incorporated information on adjacent amino acid pairs into the label design.

There are 400 possible ordered pairs of adjacent amino acids, because there are 20 choices for each position. These 400 labels were added to the original 20 amino acid labels. The order matters: for example, AE and EA are different sequence elements.



Label	Data number
AE	178
CK	178
EM	178
HW	178
MT	178
TH	178
WC	178
Ala(A)	178
Cys(C)	178
Glu(E)	178
His(H)	178
Lys(K)	178
Met(M)	178
Thr(T)	178
Trp(W)	178

Label	Data number
R	22
F	22
P	22
Y	22
RV	21
V	21
H	19
D	16
YI	15
I	15
PF	11
DR	9
HP	9
IH	9
VY	9

Label	Data number
C	176
Q	176
K	176
S	175
E	174
CK	173
QW	173
ES	173
HQ	173
ST	173
TH	173
WC	173
H	173
T	173

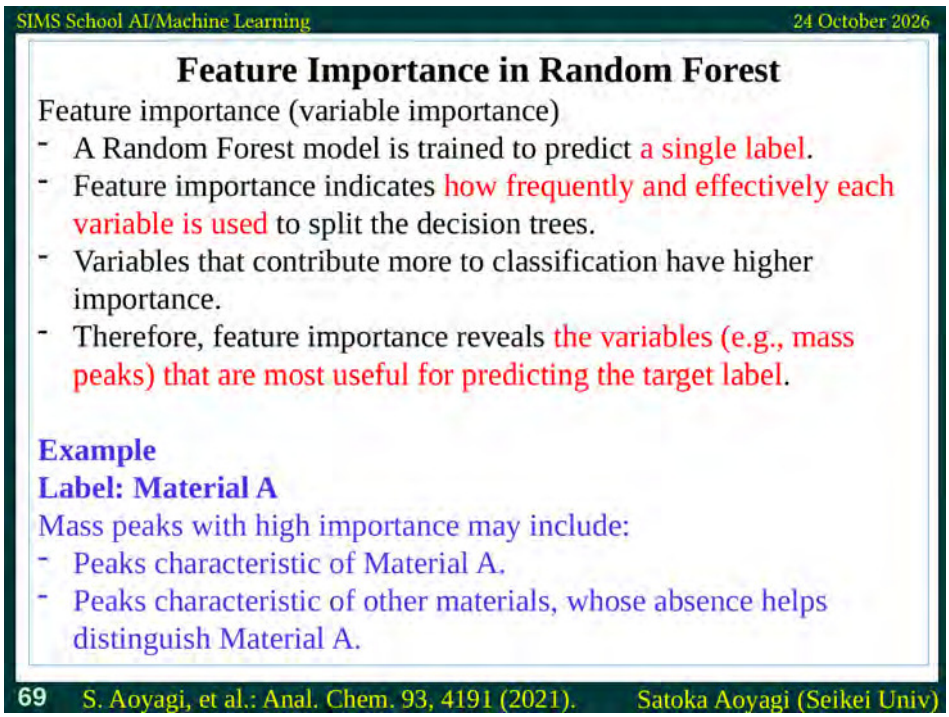
Label	Data number
W	173
YG	172
M	172
L	141
AD	102
NK	102
DP	102
GL	102
LM	102
KF	102
FY	102
PN	102
A	102
N	18

Label	Data number
RV	23
R	23
F	23
P	23
Y	23
V	23
H	20
D	17
YI	14
I	14
RR	6
GR	6
GG	6
IH	6
MR	6
FM	6
PE	6
VG	6
E	6

Using these labels, we correctly reconstructed the sequences of the peptides included in the training data in this analysis.

Even for peptides not included in the training dataset, partial amino acid sequences could be predicted.

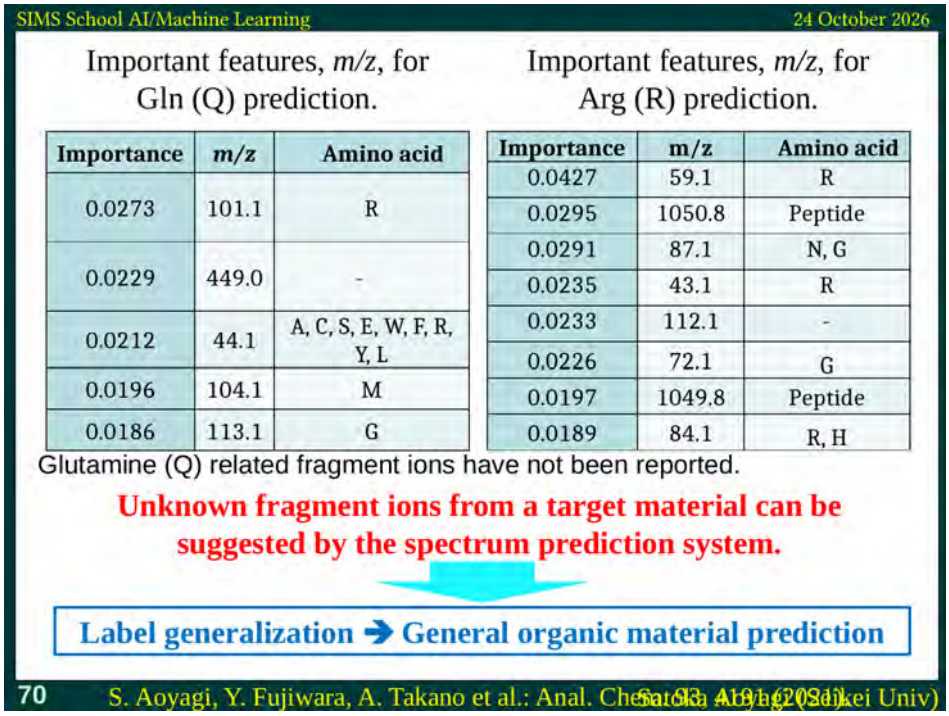
The tables compare interpolation and extrapolation results. We should distinguish a set of predicted adjacent pairs from an unambiguous reconstruction of a complete sequence. Nevertheless, these predictions provide more structural information than amino acid presence alone.



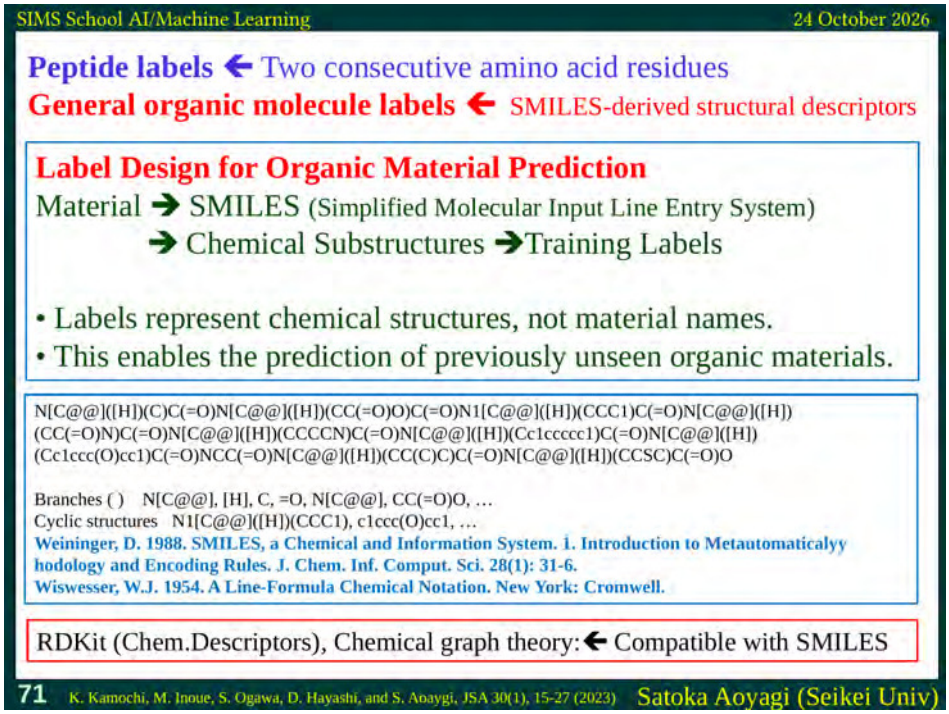
RF importance can also highlight mass peaks associated with a particular chemical structure, such as an amino acid.

These peaks may characterize the target material or another material whose absence helps distinguish the target.

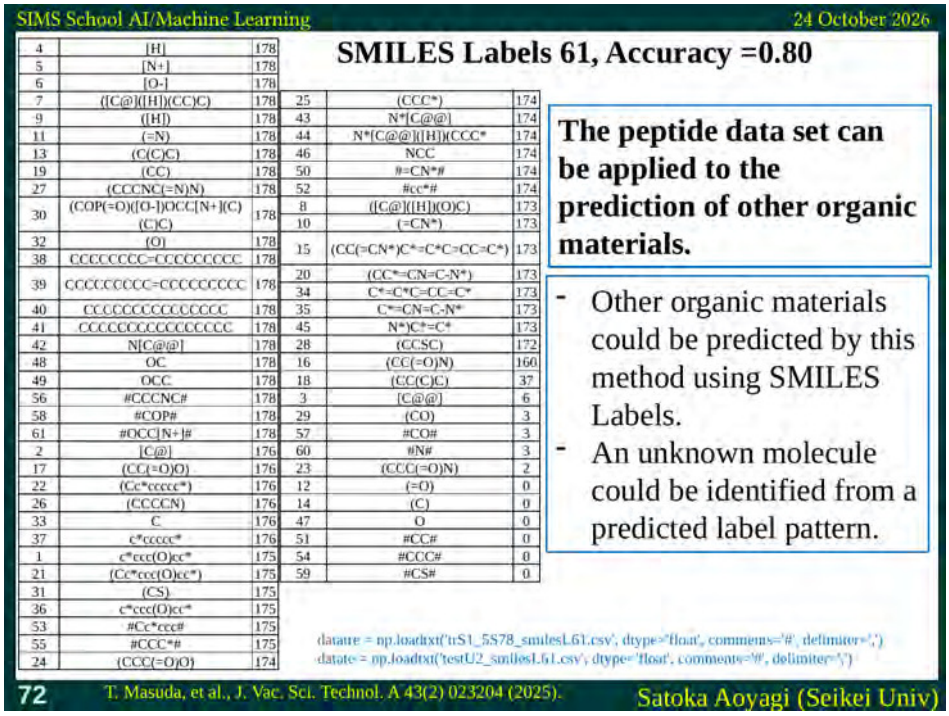
Importance guides further investigation; it does not prove a fragmentation mechanism.



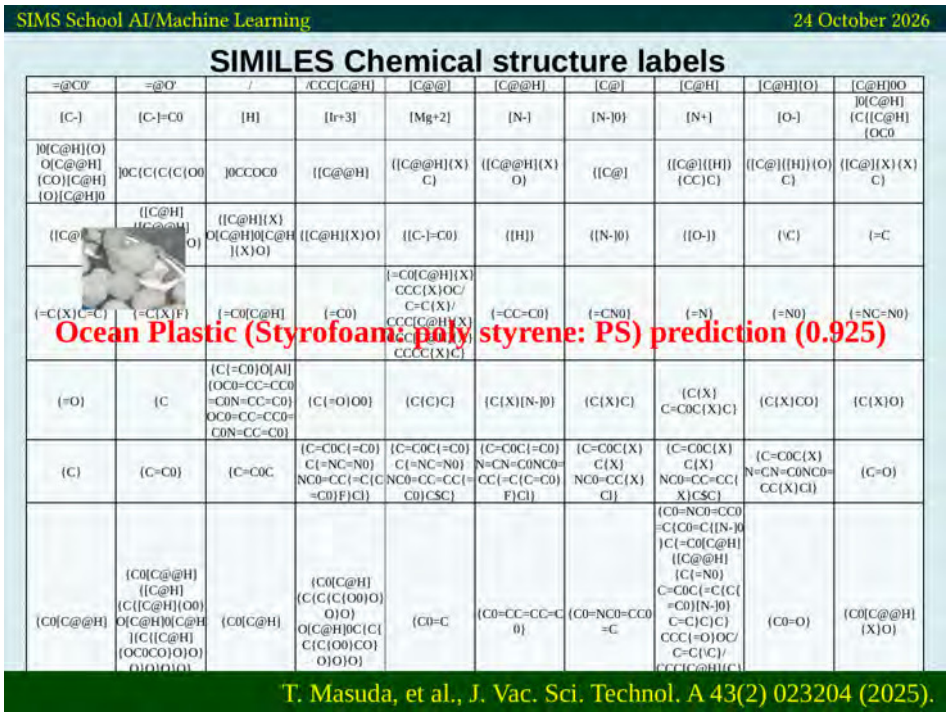
These tables list peaks useful for predicting glutamine and arginine labels. The glutamine results suggest candidates for further examination where established fragment assignments were limited. A peak can support a label prediction without originating exclusively from that residue, so chemical assignment and independent checks remain necessary.



We extend structural label design from peptide residues to general organic molecules. Each molecule is represented by a SMILES string, from which smaller structural elements are extracted as labels. The model predicts these structural elements from a spectrum. The predicted pattern can then be compared with candidate molecules, rather than being restricted to a numerical ID for each training material.



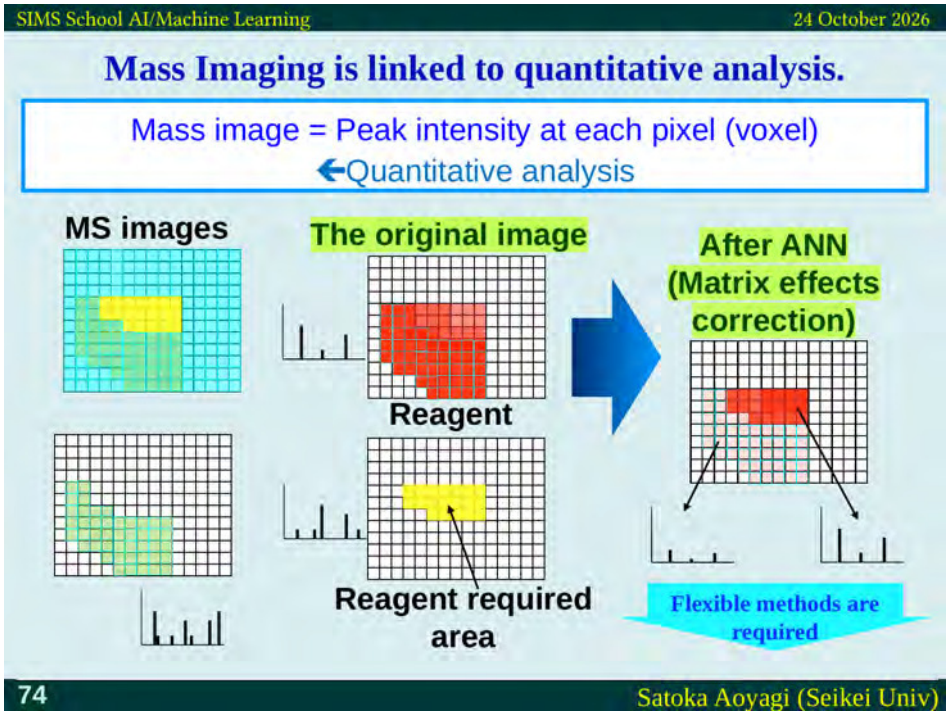
This slide shows an example using 61 SMILES-derived labels. The predicted label pattern can be compared with the structural elements of candidate molecules. A reported accuracy describes the performance in the particular evaluation shown here. The important idea is that chemical structures shared among molecules can provide information beyond a fixed list of material names. Whether a previously unseen molecule can be identified depends on the coverage and discriminating power of those labels.



Based on the SMILES labels, unknown ocean plastic samples were predicted as polystyrene with a high level of label agreement. These ocean plastic samples were also confirmed to be polystyrene using TOF-SIMS MSMS.

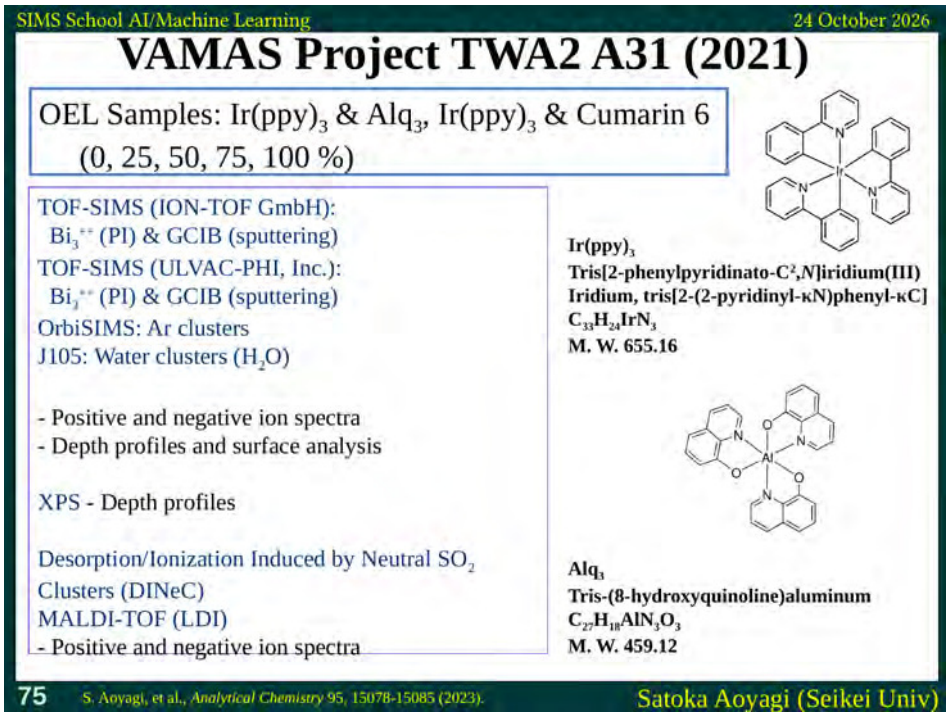
The displayed value of 0.925 represents the label agreement reported for this example. Agreement with predicted chemical-structure labels supports a candidate identification, while the MS/MS measurement provides an independent chemical check.

Please let us know if you are interested in our material prediction system, which is based on RF and SMILEs. We can share our programmes with you.



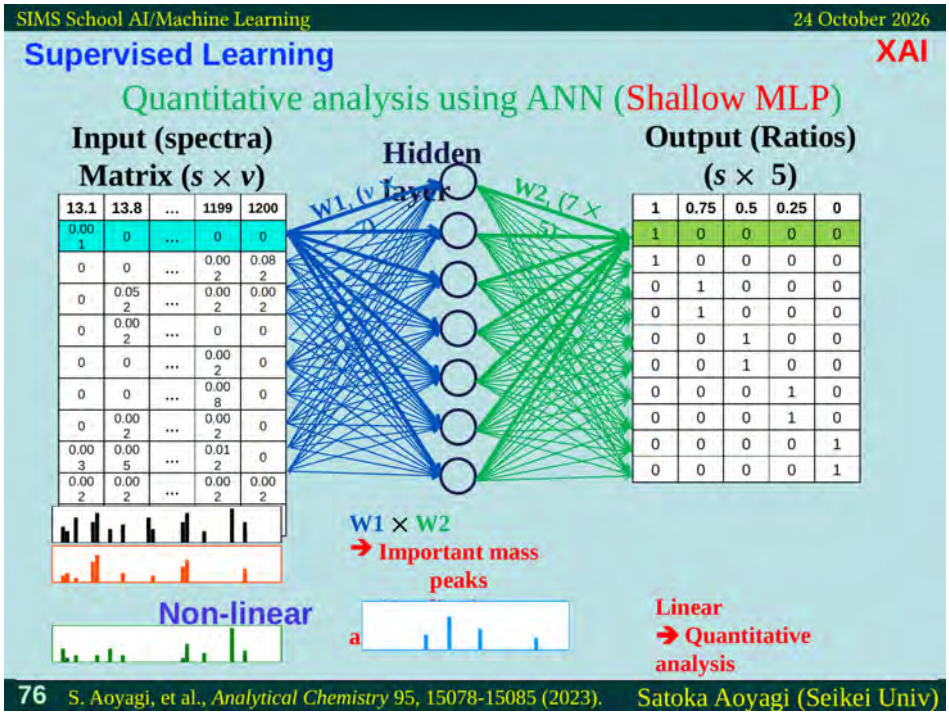
In mass imaging, peak intensity can change with both the amount of a substance and its chemical environment, which can cause matrix effects. This schematic shows how matrix effects can enhance or suppress the intensity of a particular secondary-ion mass peak. As a result, the image based on that peak may not accurately represent the actual distribution of the substance.

Obtaining accurate mass images requires reliable quantitative analysis that accounts for matrix effects.



Two sets of binary OEL mixture samples were analysed by ToF-SIMS with Bi cluster and water cluster ion beams, and OrbiSIMS with an Ar cluster ion beam, as well as laser desorption/ionisation (LDI), and desorption/ionisation induced by neutral SO2 clusters (DINeC).

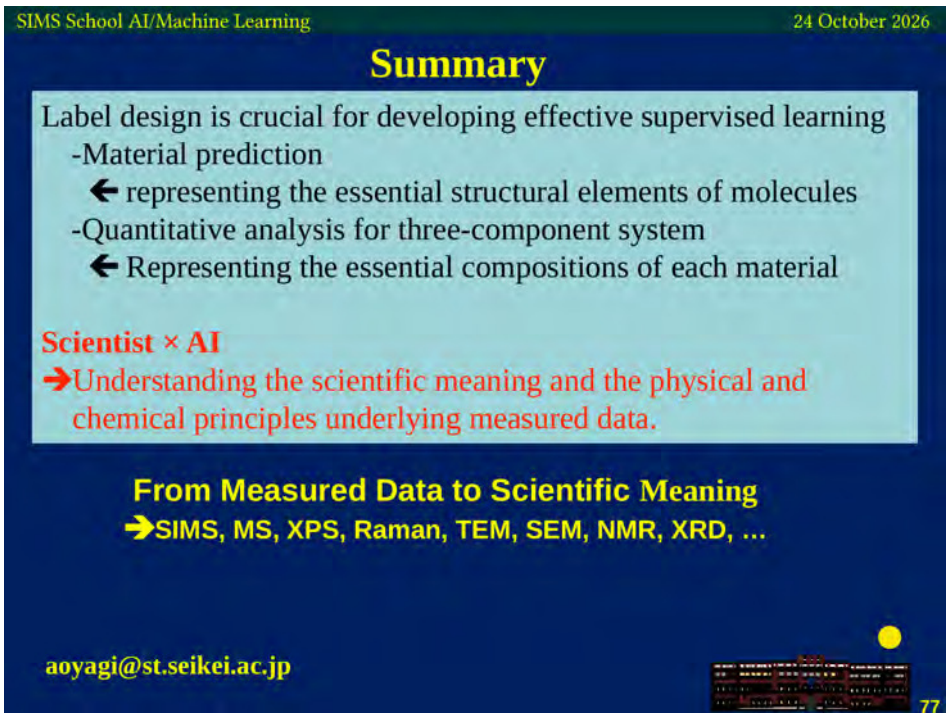
XPS was also employed for reference.



We have developed a shallow multilayer perceptron (MLP)-based system for quantitative analysis from SIMS spectra.

This shallow MLP can predict sample compositions and identify the mass peaks contributing to each label representing the composition.

Please let us know if you are interested in quantitative analysis system, which is based on ANN. We can share our programmes with you.



From Measured Data to Scientific Meaning

Measured Data

↓

Knowledge Representation — encode scientific knowledge

↓

Machine Learning — extract patterns

↓

Explainable AI — identify what drives the result

↓

Scientific Interpretation — connect results with physics/chemistry

↓

Validation — test alternative explanations

↓

Scientific Knowledge / Discovery

AI should not only provide answers. It should help us understand why the measured data contain those answers.

Acknowledgements

Versailles Project on Advanced Materials and Standards (VAMAS)

- Technical Working Areas (TWA) 2, 2014: sub-project A3(g) static SIMS interlaboratory study: organic depth profiling of mixed materials - protocol for analysis.

- **VAMAS TWA2, A26 (2019-21) AI/Machine learning application to Static-SIMS spectrum data analysis.**
- **VAMAS TWA2, A30 (2021-) Evaluation of Matrix Effect Correction and Investigation of Interface Determination.**

Thank you for your kind attention!

