WHITE PAPER

Alchemite™ explained

How the Alchemite™ algorithm empowers machine learning solutions for R&D

intellegens.com | info@intellegens.com
Intellegens, The Studio, Chesterton Mill, Cambridge, CB4 3NP, UK

intellegens

Executive Summary

Machine learning is increasingly applied to accelerate R&D in sectors such as materials, chemicals, life sciences, and formulated products. Alchemite™, developed originally at the University of Cambridge and, since 2017, at Intellegens, is an algorithm tuned for the challenges found in these research areas. Notably, Alchemite™ works well with real experimental and process data, which is typically sparse and noisy, on which other machine learning methods fail. Another strength is in handling the uncertainty calculations that can be vital, for example, when prioritising experimental work. This white paper lifts the lid on some of these features, providing methodological and mathematical background for scientists interested in what underlies the unique capabilities of the Alchemite™ software.

Introducing Alchemite™ machine learning

What is machine learning?

Machine learning (ML) is a branch of artificial intelligence (AI) that applies statistical algorithms to learn from data, creating models that are then used for tasks such as taking a new set of inputs for the system to predict the outputs. In the experimental sciences, this can guide and focus experimental programs, saving time and money. Figure 1 is a schematic illustratation of supervised machine learning, which learns from labelled data that describes the system's inputs and outputs.

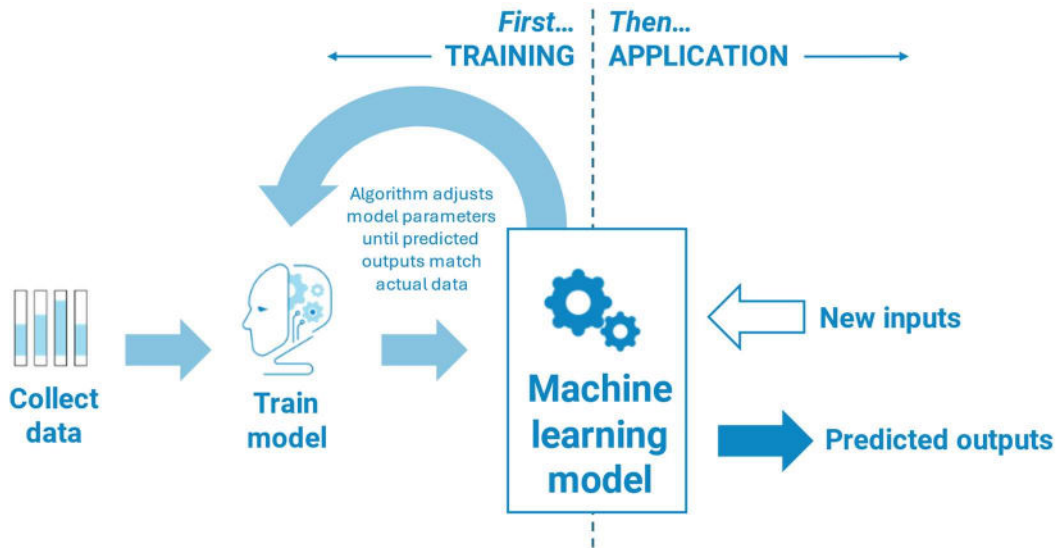

Figure 1. Schematic illustration of supervised machine learning

The ML model is a mathematical representation of the process by which inputs combine to produce outputs. A machine learning study typically starts with a generic model and 'trains' it. The model is given real input data and its parameters are automatically and iteratively adjusted until its prediction for the outputs is as close as possible to the outputs in the training data. The trained model can then be used to predict results for unseen data. We might use these predictions to propose new experiments, and the results of those experiments could then be used to re-train the model in a continuous improvement cycle.

Machine learning enables us to understand and model processes with no need for explicit instructions from an operator. It can find and exploit relationships in data that human analysts would miss and avoid biases in interpreting data. In R&D, however, it is typically not used to replace expert analysis, but as a complement to scientific expertise – deepening insights and focusing work on the likeliest routes to experimental success.

What is Alchemite™?

Alchemite™ is a proprietary supervised machine learning algorithm originally developed in the University of Cambridge and, since 2017, further developed and applied by Intellegens. It uses random forests and a variety of specialist algorithmic and computational adaptations that make it particularly suitable for problems based on experimental or process data. This white paper reviews the most important of these features.

An R&D focus

Alchemite™ was originally validated in the design of aerospace alloys at Rolls-Royce [1]. Problems in materials research are a continued focus. Alchemite™ can take composition and processing parameters as input to predict physical properties, as shown in Figure 2.

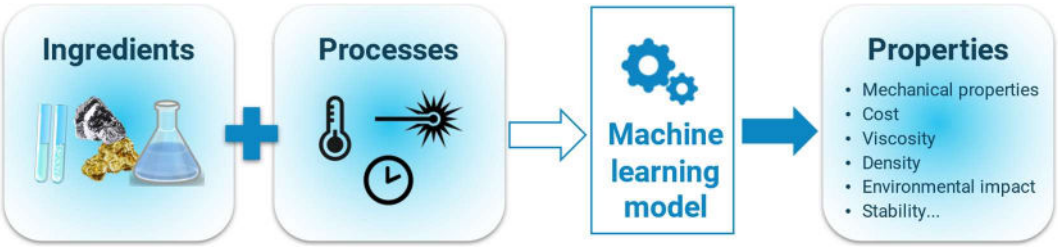

Figure 2. Alchemite™ takes the composition and processing parameters through a model to predict the physical properties of a material or formulation.

Such an approach generalises well to many areas of scientific R&D – for example, development of formulated products, study of chemical and biological processes, and optimization of advanced manufacturing. These applications have many things in common, including the frequency of complex optimization problems and the need to work with sparse, noisy experimental and process data. As we shall see, Alchemite™ is ideal for meeting these challenges. The rest of this paper lifts the lid on some of its underlying features by following the journey of data through Alchemite™, from ingestion through to predictions.

Data

Machine learning is built on data. For Alchemite™, training data is straightforward to prepare: a table where each row represents a different experiment (e.g., for a different material, chemical, or formulation) and the columns represent the values of the various input and output quantities. An exemplar dataset for steels is shown in Figure 3.

Ni	Mo	Al	Ti	Heat treatment/C	YS/MPa	UTS/MPa	Elong/%
15.1	5.1	0.9	0.7	982	1619.2	1722.5	9
12.1	2.8	0.2	0.2	816	1167.2	1220.9	
12.2	3.0	0.3	0.2			1249.8	17
12.2	3.0	0.4	0.2	816		1318.1	17.5
12.1	3.1	0.6	0.2	816	1390.4		15
12.1	3.2	0.4	0.2			1267.1	
11.5	3.0	0.7	0.2	816	1451.7	1494.4	
11.6	3.0	0.6	0.2	816	1437.3		16
13.0	2.9	0.5	0.2	816	1404.2		15
11.8	3.0	0.4	0.2	816	1324.3		16
12.2	3.0	0.3	0.2	816	1309.8	1345.6	15
12.1	3.0	0.2	0.2	816			

Figure 3. Example dataset for steels. Each row is a different material, with the columns being design variables and measured properties.

Composition information is present, with concentrations of four elements in the first four columns. The fifth column contains heat treatment temperature, also regarded as an input in this material production process. The properties: yield strength, ultimate tensile strength, and elongation are in the final three columns. These are the measured properties of the material – in machine learning terms we regard these as outputs. Note that the data is *sparse* (has gaps) since, as is typical in real experimental programs, not every property was measured in every test. Provided that data can be formatted in this way, it is ready for machine learning.

Model validation

Ni	Mo	Al	Ti	Heat treatment/C	YS/MPa	UTS/MPa	Elong/%
15.1	5.1	0.9	0.7	982	1619.2	1722.5	9
12.1	2.8	0.2	0.2	816	1167.2	1220.9	
12.2	3.0	0.3	0.2			1249.8	17
12.2	3.0	0.4	0.2	816		1318.1	17.5
12.1	3.1	0.6	0.2	816	1390.4		15
12.1	3.2	0.4	0.2			1267.1	

Ni	Mo	Al	Ti	Heat treatment/C	YS/MPa	UTS/MPa	Elong/%
11.5	3.0	0.7	0.2	816	1451.7	1494.4	
11.6	3.0	0.6	0.2	816	1437.3		16
13.0	2.9	0.5	0.2	816	1404.2		15
11.8	3.0	0.4	0.2	816	1324.3		16
12.2	3.0	0.3	0.2	816	1309.8	1345.6	15
12.1	3.0	0.2	0.2	816			

Figure 4. Separated training (upper) and blind validation data (lower). The training data is used to set up the machine learning, whereas the blind validation data is used to validate the accuracy of the model.

The formal approach to validate the model is to split the dataset (Figure 4): with part being used to train the model, and the remainder being held unseen to mimic fresh experiments, against which we formally test the accuracy. Typically, 80% of the data is used for training and

20% held back for validation. In a process known as cross-validation, this 20% validation data can be cycled through the entire dataset, so that all data is used for blind validation.

Next, we discuss the two separate metrics used in validation, before combining them into a single metric that we can optimize when selecting the model parameters.

Metric for accuracy

The metric for the accuracy of predictions is the coefficient of determination, chosen as it is the simplest metric independent of both the origin and scaling of the data:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - p_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}$$

where y_i is the i^{th} data entry of N, and p_i is the model prediction of the i^{th} value. The limit $R^2 = 1$ represents perfect predictions; $R^2 = 0$ are poor predictions that simply estimate the average.

Metric for uncertainty

Understanding the uncertainty in our predictions is just as important as the predicted value itself, so we introduce a metric to gauge the quality of Alchemite's uncertainty estimates. The principle is that the typical prediction should be within its uncertainty of the true value and, indeed, should follow a normal distribution. We therefore compare the distribution of:

$$\epsilon_i = \frac{y_i - p_i}{\sqrt{\sigma_{\mathrm{y},i}^2 + \sigma_{\mathrm{p},i}^2}}$$

to a normal distribution of mean 0 and standard deviation 1, where y_i is the i^{th} data entry with uncertainty $\sigma_{\mathrm{y},i}$, p_i is the model prediction of the i^{th} value with uncertainty $\sigma_{\mathrm{p},i}$. If $\sigma_{\mathrm{y},i}$ is unknown, then Alchemite™ can estimate it from internal parameters.

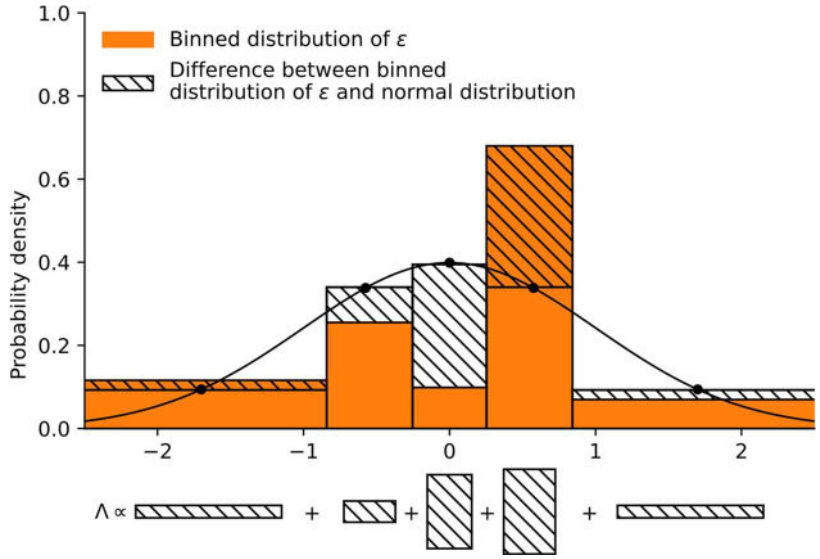

Figure 5. Distribution of ϵ binned and compared to the desired normal distribution. Reproduced from [2].

To formally undertake the comparison, we bin the ϵ_i distribution into $\sqrt{N}$ bins chosen so that the bins should have equal number of entries. This distribution is shown in Figure 5.

We then calculate our error in uncertainties metric that sums the area difference between the achieved distributions of errors to the reference normal distribution:

$$\Lambda = \frac{1}{2N}\sum_{i=1}^{\sqrt{N}} |n_i - \sqrt{N}|$$

normalized so that $\Lambda = 0$ is excellent estimation of the uncertainties, and $\Lambda = 1$ means poor estimation of uncertainties. Further details are discussed in [3].

Combined metric

So that both quality of predictions and uncertainties are considered on an equal footing we ensure that the Alchemite™ hyperparameter optimization maximizes the quantity:

$$R^2 - \frac{2(1-R^2)\Lambda}{\sqrt{1/2 - 1/\pi}}$$

The metric is carefully selected so that when in the limit of perfect predictions, R^2=1, the combined metric approaches R^2 as here machine learning bootstrapping methodology must necessarily give correct uncertainties with $\Lambda = 0$ so this metric can be ignored. As predictions worsen in quality and R^2 decreases, the combined metric places increasing weight on Λ, so that the uncertainties can be relied upon to focus on only the most reliable predictions. Further details of the combined metric are discussed in [2].

Overcoming difficult data

Experimental data is never a clean set of simple numbers. It can be sparse, noisy, or contain different kinds of information – for example, images. Below we detail how Alchemite™ extracts as much information as possible from different types of difficult data.

Sparsity

We are often interested in multiple properties. For example, for a material to be commercially viable we require that a material is both strong, lightweight, and affordable. However, not all properties are always measured, rendering the dataset sparse. Such a dataset was shown in Figure 3.

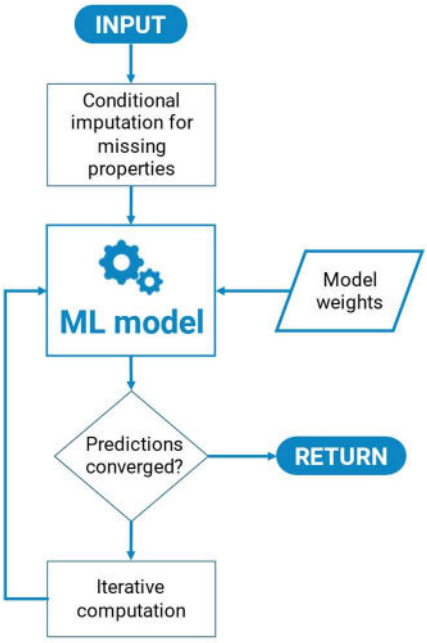

Figure 6: Algorithm flowchart for imputation of sparse data. Reproduced from [4].

Alchemite™ has a unique ability to extract information from a dataset with missing values, uncovering and then exploiting relationships between properties to impute values. This is in contrast to standard machine learning tools, which cannot usually handle sparse data with ease. Alchemite™ achieves this by judiciously cycling around the dataset, imputing values until the model reaches self-consistency, as shown in the workflow in Figure 6. Further details can be found in [5].

Noise

Another common feature of experimental data is that measurements are susceptible to random error. Alchemite™ constructs an ensemble of models, each trained differently to reflect variation in the training data, and will self-average over values in the training data to deliver a more accurate prediction:

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Alchemite™ also takes advantage of the variation in the models to estimate the uncertainty:

$$\frac{1}{N^{3/2}}\sum_{i=1}^{N} (y_i - \overline{y})^2$$

This delivers the following performance for fitting of a straight line, and appropriate increase in uncertainty for extrapolation.

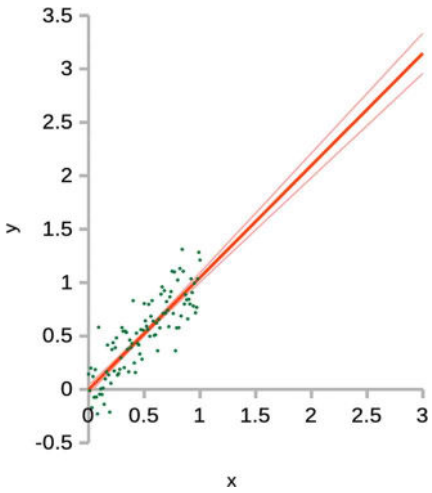

Figure 7: Uncertainty in fitting a straight line to noisy training data. The green points are training data, the red line is the model prediction, and the pink lines the one standard deviation uncertainty that grow with extrapolation.

Additional consideration is required for variables that Alchemite™ deems unnecessary to build the model as, if these are taken outside of their known regime and extrapolated, model behaviour is unknown. Alchemite™ therefore has a built-in uncertainty extrapolation for unused variables to make sure the final uncertainty reflects the true uncertainty in the model.

Alchemite™ distinguishes between uncertainty in the model (typically due to extrapolating beyond the range of training data) versus noise in the experimental data. The direct estimate of the uncertainty is the simply the uncertainty in the model σ_{model}, and so predictions should

be directly compared to precise experimental data. If the experimental data also carries uncertainty, then this should be included in the comparison of the model to the measured value:

$$\sqrt{\sigma_{\text{model}}^2 + \sigma_{\text{experiment}}^2}$$

Should the experiments being compared to be performed to the same level of accuracy as those in the training data, the quantity $\sigma_{\text{experiment}}$ can be estimated by Alchemite™ using internal parameters.

Use of uncertainty

As we have seen, Alchemite™ can estimate not only the prediction but also the uncertainty in it. This uncertainty is useful not only when validating the model, it can also to inform us of the robustness of predicted values when the model is being applied, and guide decision making in several useful ways.

Optimization and design of experiments

When optimizing a material, chemical, or formulation we are interested in the probability that it actually fulfils our target criteria. This probability can be calculated using the area under the probability density in Figure 8.

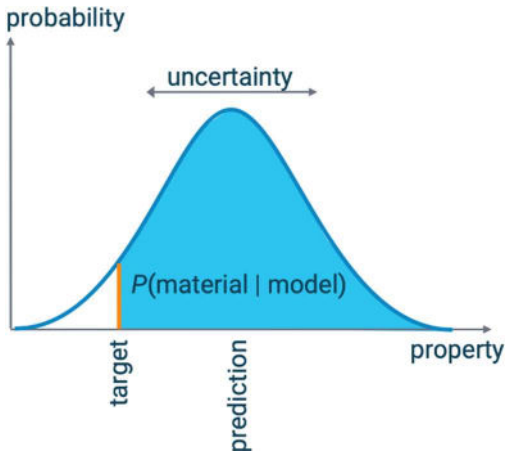

Figure 8: Probability density of a prediction (blue) relative to the target (orange). The shaded area shows the probability of fulfilling the target given the model, $P(model \mid material)$.

However, this finds $P(\text{material} \mid \text{model})$, that is the probability that the formulation will fulfil the target assuming that the machine learning model is true. To calculate the quantity of interest, $P(\text{material})$, the probability that the material will fulfil the target we need to use Bayes' theorem:

$$P(\text{material}) = \frac{P(\text{material} \mid \text{model})\, P(\text{model})}{P(\text{model} \mid \text{material})}$$

where we have expressions for $P(\text{model})$ and $P(\text{model} \mid \text{material})$ that we have derived using internal knowledge of our Alchemite™ model.

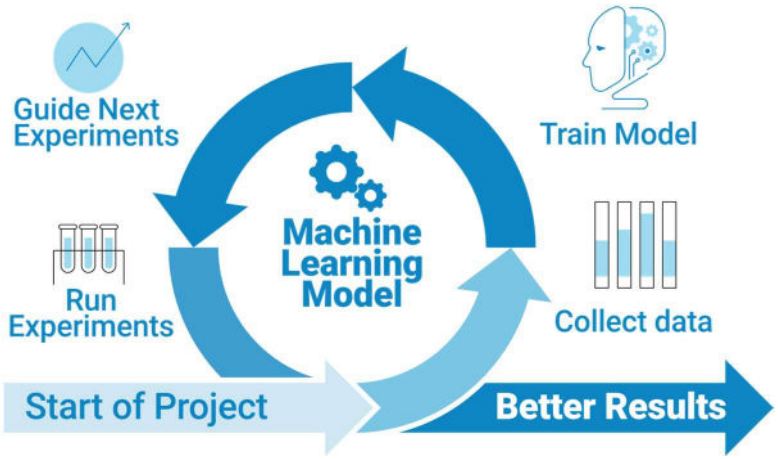

Figure 9. A schematic of the adaptive experimental design approach.

Design of experiments focuses on gathering additional data that can be later used to retrain and improve the model. The cycle is shown in Figure 9. Data is most effectively gathered where the model is uncertain, though weighted by $P(\text{model} \mid \text{material})$ to ensure that we collect data for formulations that hold promise. The correct calculation of $P(\text{material})$ is crucial to ensure the efficacy of our optimization and also the design of experiments facility, increasing the efficiency of design of experiments by a factor of ×3.

Predictions

The calculation of $P(\text{material})$ is also important to allow the robustness of individual predictions to be understood. This allows us to focus on those predictions most likely to be accurate. As shown in Figure 10, by ejecting those results with largest uncertainty we can deliver a significant average improvement in accuracy.

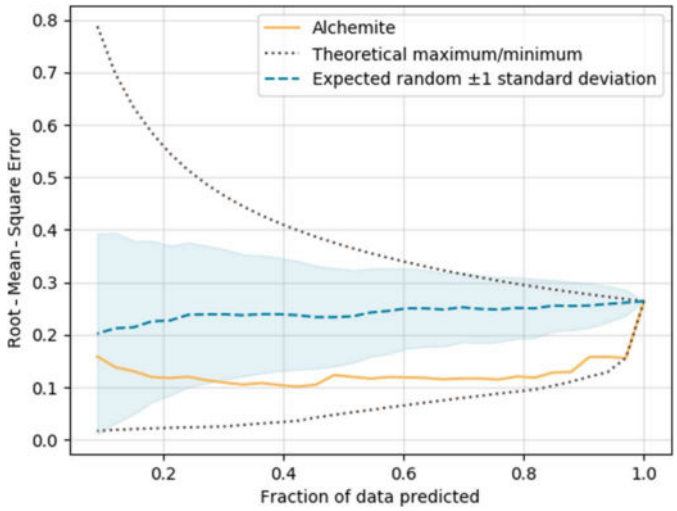

Figure 10: Improved accuracy (blue dashed line) when predicting the least uncertain results. Reproduced from [6]

Conclusion – Alchemite™ unleashed

Alchemite™ has many bespoke algorithmic features allowing it to deliver accurate predictions on difficult data. This is backed up by a large library of real-world use cases spanning many industries, a selection of which are publicly available as case studies [7], peer reviewed papers [8], or white papers [9] on the Intellegens website. With the bespoke algorithms behind Alchemite™ now proven to be effective across the experimental sciences, it provides a solid foundation for the widespread adoption of machine learning.

There are two main ways in which you can apply the Alchemite™ algorithm:

- **Using the targeted apps of Alchemite™ Suite**, which provide easy-to-use browser-based interfaces (Figure 11) enabling you to complete key R&D tasks quickly and easily, drawing on the underlying machine learning algorithm. Several of the features examined in this paper are vital to enabling this app-based approach. Handling of sparse data, for example, enables users to jump right into an app and generate meaningful results without the need to pre-process data, while accurate uncertainty quantification is essential, for example, to enabling adaptive Design of Experiments.
- **Using the Alchemite™ Architect API**, which enables you to code and script around the Alchemite™ algorithm, integrating it into your research tools and projects.

Contact us to start applying Alchemite™ in your research immediately.

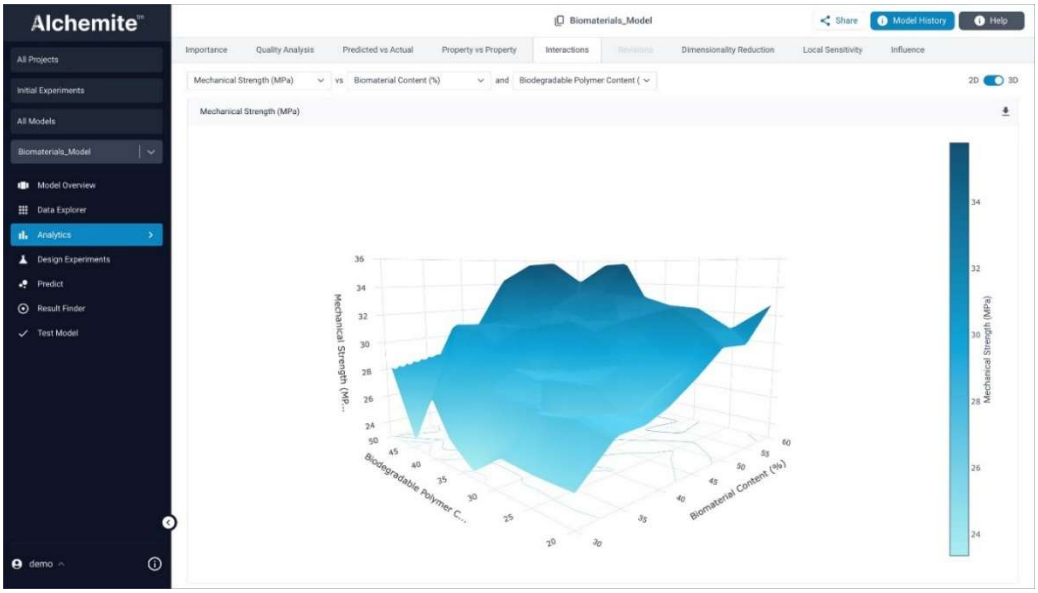

Figure 11. Alchemite™ Innovator – an app that enables to build models, apply them in predictions, and view advanced graphical visualizations such as this comparison of three material properties.

References

1. B.D. Conduit, N.G. Jones, H.J. Stone & G.J. Conduit, Design of a nickel-base superalloy using a neural network, *Materials & Design* **131**, 358 (2017)

2. Zviazhynski, B. *Machine learning to extract information from noise: application to concrete and cancer detection* [Apollo - University of Cambridge Repository] (2024) https://doi.org/10.17863/CAM.113245

3. B. Zviazhynski & G.J. Conduit, Unveil the unseen: exploit information hidden in noise, *Applied Intelligence* (2022)

4. S.Y. Mahmoud, B.W.J. Irwin, D. Chekmarev, S. Vyas, J. Kattas, T.M. Whitehead, T. Mansley, J. Bikker, G.J. Conduit & M.D. Segall, Imputation of Sensory Properties Using Deep Learning, *Journal of Computer-Aided Molecular Design* **35**, 1125 (2021)

5. B.D. Conduit, T. Illston, S. Baker, D. Vadegadde Duggappa, S. Harding, H.J. Stone & G.J. Conduit, Probabilistic neural network identification of an alloy for direct laser deposition*, Materials & Design* **168**, 107644 (2019)

6. B.W.J. Irwin, J. Levell, T.M. Whitehead, M.D. Segall & G.J. Conduit, Practical Applications of Deep Learning to Impute Heterogeneous Drug Discovery Data, *Journal of Chemical Information and Modeling* **60**, 2848 (2020)

7. Intellegens case studies, https://intellegens.com/casestudies

8. Intellegens publications, https://intellegens.com/article-type/papers/

9. Intellegens white papers, https://intellegens.com/article-type/white-papers/

About Intellegens

Our vision is that machine learning will drive innovation and deliver value wherever data is used in R&D. We aim for best-in-class easy-to-use machine learning software for data analysis in chemicals, materials, life science, and manufacturing. Our Alchemite™ technology originated at the University of Cambridge and development is on-going at Intellegens, in close collaboration with our growing community of customer organizations. These represent sectors including additive manufacturing, aerospace, alloys, batteries, biopharmaceuticals, ceramics, chemical processes, composites, consumer products, cosmetics, drug discovery, energy, food and beverage, formulated products, paints, plastics, and printing technology.

www.intellegens.com | info@intellegens.com