

ICE Machine Learning Tool

Introduction

The ICE machine learning tool allows you to use different data types to build a model to predict a value for an in vivo toxicological endpoint. The spring 2019 release of ICE allows you to predict:

- Mouse and human skin sensitization hazard
- Estrogenic activity in the rodent uterotrophic assay

The machine learning tool used in ICE is discussed in detail in [Appendix 1](#). A standalone version in the R programming language is available for [download](#) and has additional outputs and options not found in the ICE version. This User Guide focuses on the ICE implementation of the tool.

Building a Machine Learning Workflow

Figure 1 shows the ICE machine learning tool Input view. You can toggle between Input view and Results view by clicking tabs on the left side of the screen. The tool window defaults to Input view when it is first opened.

The screenshot displays the 'ICE machine learning tool Input view' interface. At the top, the header identifies the 'National Toxicology Program' and 'Integrated Chemical Environment'. A search bar is located in the top right corner. Below the header, a navigation bar includes tabs for 'Input', 'Results', and 'Machine Learning'. The 'Input' tab is currently selected. The main content area is titled 'Machine Learning Tool Input' and includes a 'Run Tool' button. Under 'Prediction Type', 'Classification' is selected. The 'Select Machine Learning Methods' section lists 'rpart', 'knn', 'svmRadial', 'cforest', and 'pls'. The 'Chemicals to use' section has buttons for 'Select Chemicals' and 'Select Chemical Quick List'. A text area for 'Enter one CASRN per line.' is provided at the bottom right. A 'BACK TO TOP' link is visible in the bottom left corner.

Figure 1. ICE machine learning tool Input view.

Select Prediction Type and Endpoints

In Input view, use the dropdown list to select either “Classification” or “Regression” as shown in **Figure 2**.

The screenshot displays the NTP ICE Machine Learning tool interface. The 'Prediction Type' dropdown is set to 'Regression'. The 'Human Potency Call' radio button is selected. The 'Select Machine Learning Methods' section shows 'Human Potency HMT, NOEL' as the selected method. The interface includes a 'Run Tool' button, a 'Select Data for Model Building' button, and a 'Select Chemicals' button. The 'Enter one CASRN per line.' text is visible at the bottom.

Figure 2. Options for Classification and Regression models.

Classification models return a binary prediction, such as toxic/non-toxic or positive/negative. Endpoint options for classification models currently include:

- LLNA call (skin sensitization hazard as predicted by the mouse local lymph node assay or LLNA)
- Human potency call (skin sensitization hazard as predicted by human skin sensitization assays)
- Uterotrophic call (estrogenic activity in the rodent uterotrophic assay)

Regression models return a numerical prediction. Endpoint options for regression models currently include:

- Human skin sensitization potency endpoints
 - Human maximization test no-effect level (NOEL)
 - Human maximization test lowest-effect level (LOEL)
 - Human repeat insult patch test NOEL
 - Human repeat insult patch test LOEL

- Uterotrophic lowest effect level (LOEL)

Select Assays and Endpoints

Click the “Select Data for Model Building” button to specify the assay endpoints for the predictive model. This will open a dialog box for assay selection.

- Use the dropdown list within the dialog box to select an assay and click the checkboxes that appear to select endpoints from that assay.
- Click the dropdown list again to add endpoints from a different assay.
- Click “Finished Data Selection” when you are done.

All of the selected assays and endpoints will appear in a table under the “Select Data for Model Building” button. These steps are shown in **Figures 3, 4, and 5**.

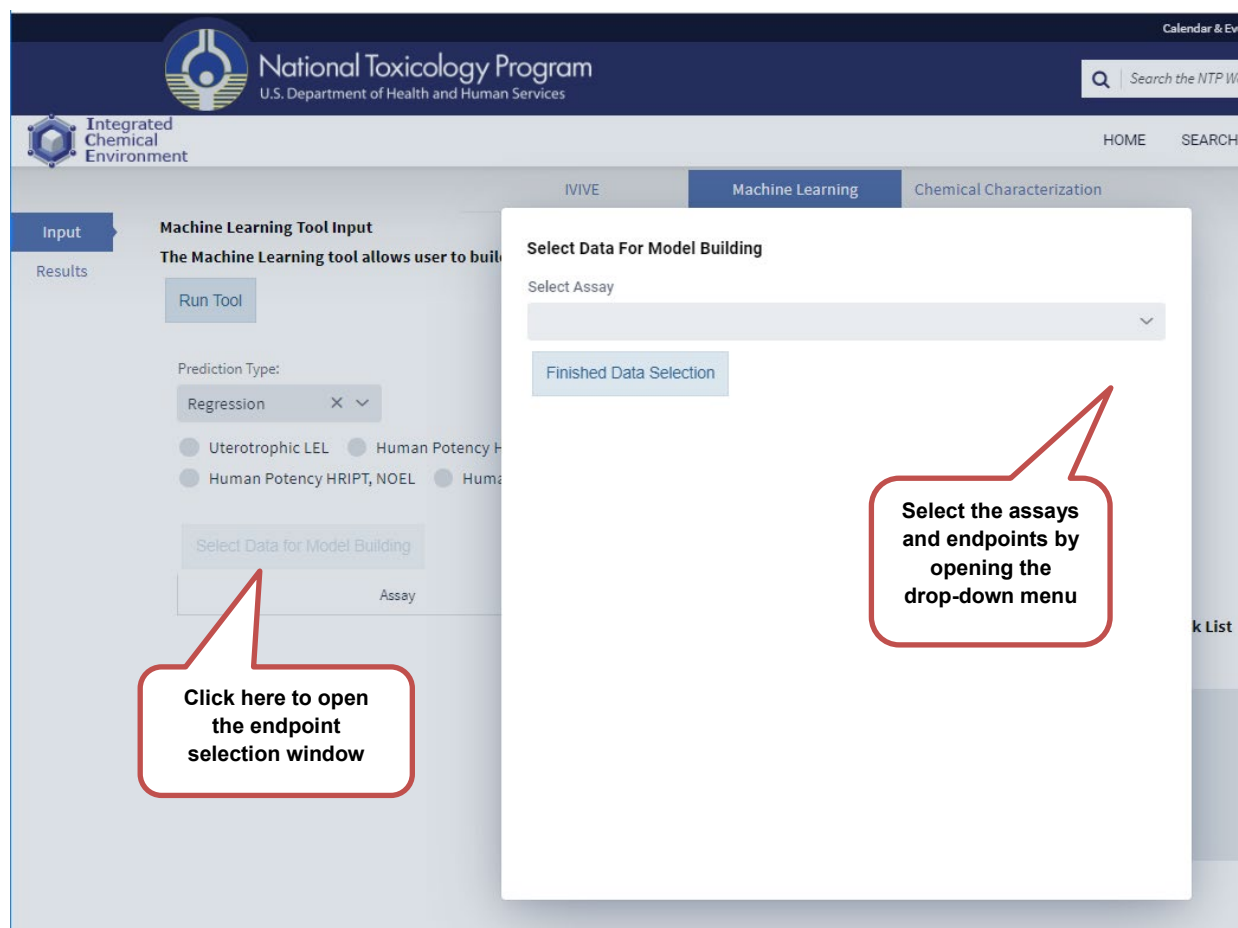


Figure 3. Use the dropdown list in the dialog box to select assays for model building.

Select Data For Model Building

Select Assay

PhysChem Properties ✕ ▼

☒ AOH
 ☐ BP
 ☐ CLint
 ☐ fu
 ☐ HL
 ☒ KOA

☐ BCF
 ☒ LogD, pH 5.5
 ☐ LogD, pH 7.4
 ☐ LogP

☒ MP
 ☐ MW, Molecular Weight
 ☒ pKa_ionization

☐ pKa_Acidic
 ☐ pKa_Basic
 ☒ VP
 ☐ WS, Water Solubility

Finished Data Selection

Use the drop-down list to view more assays

Click this button once you are done

Figure 4. Select assays for model building. Once done, click on “Finished Data Selection”.






Select Data for Model Building		
	Assay	Endpoint
	PhysChem Properties	VP
	PhysChem Properties	MP
	PhysChem Properties	pKa_ionization
	PhysChem Properties	LogD, pH 5.5
	PhysChem Properties	KOA

Figure 5. Selected assays and endpoints are displayed in a table in Input view.

Select Machine Learning Methods

Select one or more machine learning methods from the list on the right side of the screen (**Figure 6**):

- [rpart](#): recursive partitioning and regression trees

- [knn](#): weighted k-nearest neighbors
- [svmRadial](#): support vector machines with radial basis function kernel
- [rf](#): random forest
- [pls](#): partial least squares

You can find more information about the machine learning methods in [Appendix 2](#).

National Toxicology Program
U.S. Department of Health and Human Services

Integrated Chemical Environment

HOME SEARCH TOOLS DATA ABOUT HELP

Calendar & Events News & Media Get Involved Support

Search the NTP Website SEARCH

IVIVE Machine Learning Chemical Characterization

Machine Learning Tool Input
The Machine Learning tool allows user to build simple predictive models for data exploration using ICE data.

Run Tool

Select at least one machine learning method

Select Machine Learning Methods

☐ rpart ☐ knn ☐ svmRadial
☐ cforest ☐ pls

Chemicals to use

Select Chemicals Select Chemical Quick List

Enter one CASRN per line.

Assay	Endpoint
PhysChem Properties	VP
PhysChem Properties	MP
PhysChem Properties	pKa_ionization
PhysChem Properties	LogD, pH 5.5
PhysChem Properties	KOA

Figure 6. Machine learning method selection.

Select Chemicals

Below the machine learning methods, specify chemicals to use in the model via one or both of two input methods (**Figure 7**):

1. Select one or more chemical quick lists by clicking “Select Chemicals”, which opens a dialog box with ICE [Chemical Quick Lists](#). Check the boxes in the dialog box to choose one or more chemical lists; click “Close” at the bottom of the dialog box when done.
2. Enter your own list of CASRN (one per row) in the text box below the “Select Chemicals” button.

Select one or more chemical quick lists.

Select All Deselect All

☒ AR In Vitro Agonist 2016 (R)

☒ AR In Vitro Antagonist 2016 (R)

☐ AR In Vivo Agonists 2018

☐ AR In Vivo Antagonists 2018

☐ ER In Vitro Agonist 2015 (R)

☐ ER In Vitro Agonists and Antagonists OECD 2016 (R)

☐ ER In Vivo Agonist 2015 (R)

☐ ICCVAM Cytotox Acute Oral 2006 (R)

☐ ICCVAM Eye Irritation-Corrosion 2006 (R)

☒ ICCVAM LLNA 2009 (R)

☐ ICCVAM Skin Corrosion 2004 (R)

☐ Steroidogenesis - Androgen 2018

☐ Steroidogenesis - Estrogen 2018

☒ Thyroid 2016

☐ Tox21

☐ Uterotrophic Reference List (R)

Close

Chemicals to use

Select Chemicals 4 chemical quick lists selected.

Enter one CASRN per line.

434-22-0

521-18-6

10418-03-8

71-58-9

68-23-5

57-91-0

68359-37-5

52315-07-8

17804-35-2

85-68-7

Click this button to view the chemical lists

Click this button once you are done

Enter CASRNs: one per row

Figure 7. Specify chemicals to use in the model by entering CASRNs or selecting one or more Chemical Quick Lists. “(R)” indicates a reference chemical list.

Note: to ensure there is sufficient data for building the model, all chemicals that have data in ICE for the endpoint to be predicted will be added to the input chemical list as “seed chemicals”. Once the results are generated, the user will see whether the chemicals are from the user-provided list or seed chemicals from ICE database.

Run Workflow

Once all the assays, methods, and endpoints are selected, click the "Run Workflow" button at the top of the page. At least 10 chemicals and two assay endpoints must remain after data processing that is done before running the machine learning methods.

Viewing Machine Learning Results

The window will switch to Results view, and one or two links will appear:

- **Missing CASRNs:** for a successful workflow, this file lists chemicals that were not returned by the query, including those that are in ICE but have insufficient data for the selected modeling inputs.

- **Machine Learning Results Zip File:** for a successful workflow, this contains the following files:
 - **SummaryReport.txt:** shows the summary results of the processing steps and the summary of the performance of the different machine learning methods chosen.
 - **ModelPerformanceOutPdf.pdf:** contains graphs that show the performance of all machine learning methods using the dataset. Two or more machine learning methods must be selected to generate this plot
 - **PredictionsWithTestDataOut.txt:** provides the input after cleanup to remove all the sparse entries and imputation steps, and contains the predictions made by each of your machine learning models.
 - **Variable_Importance.pdf:** shows a plot of the relative importance of the different inputs in predicting the selected endpoint.
 - **MachineLearning_input.txt:** lists inputs provided for the machine learning tool
 - **userChemicals.txt:** contains the combined list of chemical CASRNs (user-provided chemicals and seed chemicals) used for the analysis.
 - **Error.txt:** if not enough data were provided as input or remained after preprocessing to run the workflow, this will be the only file returned.

You can use the menu in the top left to return to Input view to review or change your model parameters and rerun your model. The following section describes each output in more depth. References to help with interpreting results are provided in [Appendix 3](#).

Model Performance Results

Model performance results are returned in two files:

SummaryReport.txt

This file provides the summary results of the processing steps and the summary of the performance of the different machine learning methods chosen (**Figure 8**). It includes:

- The number of chemicals (including user-provided chemicals) and assay endpoints present in the data set before processing
- The number of chemicals and assay endpoints removed from the data set during processing due to insufficient data

- The number of chemicals and assay endpoints remaining in the data set after processing, which were used to build the model
- The name of assays retained and removed (due to insufficient data)
- The percentage of user-provided chemicals used in the model

```

Data before processing had 544 chemicals and 8 assays. From those
chemicals are provided by the user and the rest are from the seed
list
Data after processing for model construction has 478 chemicals
and 6 assays
2 assays removed with insufficient data leaving 6 assays for
model building
66 chemicals removed including chemicals from the user provided
chemical list removed with insufficient data leaving 478
chemicals for model building

Assays retained:
PhysChem.Properties.BP
PhysChem.Properties.HL
PhysChem.Properties.AOH
PhysChem.Properties.pKa_ionization
PhysChem.Properties.LogP
PhysChem.Properties.LogD..pH.5.5

Assays removed:
LLNA.Max.Simulation.Index
LLNA.EC3

Percentage of user provided chemicals available for the model:
81.82%

```

Figure 8. Contents of SummaryReport.txt file showing data present before and after preprocessing.

The SummaryReport.txt file also includes statistics specific to the prediction type selected.

Classification Models

Three statistics are provided to evaluate classification models (**Figures 9 and 10**). For each statistic, a higher number indicates better performance.

- Receiver operating characteristic (ROC): a value of 0.5 indicates the model performs no better than random and a value of 1 indicates perfect predictivity

- Sensitivity (Sens): represents how well the model predicts positive outcomes (true positive rate)
- Specificity (Spec): represents how well the models predicts negative outcomes (true negative rate)
- The file also provides a confusion matrix and statistics for each machine learning method (**Figure 10**)

```
Call:
summary.diff.resamples(object = diffModels)

p-value adjustment: bonferroni
Upper diagonal: estimates of the difference
Lower diagonal: p-value for H0: difference = 0

ROC
```

	rpart	knn	svmRadial	cforest	pls
rpart		-0.17625	0.02592	-0.14392	-0.18125
knn	0.0001244		0.20217	0.03233	-0.00500
svmRadial	1.0000000	1.123e-05		-0.16983	-0.20717
cforest	0.0011419	1.0000000	0.0054338		-0.03733
pls	0.0019816	1.0000000	7.738e-05	1.0000000	

Figure 9. Results for a classification model in the SummaryReport.txt file. Under “ROC,” data above the diagonal estimate the differences in ROC for each pair of models, while data below the diagonal are the p values for the differences. In this example, the only method not statistically different at $p \leq 0.05$ from ‘rpart’ is ‘svmRadial’ ($p = 1$; column 1 of table).

```

[1] ----- rpart -----
[1] Machine Learning Method: rpart
Confusion Matrix and Statistics

      Reference
Prediction Active Inactive
Active      11      4
Inactive     4      2

      Accuracy : 0.619
      95% CI : (0.3844, 0.8189)
      No Information Rate : 0.7143
      P-Value [Acc > NIR] : 0.8843

      Kappa : 0.0667

McNemar's Test P-Value : 1.0000

      Sensitivity : 0.7333
      Specificity : 0.3333
      Pos Pred Value : 0.7333
      Neg Pred Value : 0.3333
      Prevalence : 0.7143
      Detection Rate : 0.5238
      Detection Prevalence : 0.7143
      Balanced Accuracy : 0.5333

      'Positive' Class : Active

[1] ===== SUMMARY: rpart =====
      Accuracy      Kappa  AccuracyLower  AccuracyUpper  AccuracyNull  AccuracyPValue  McNemarPValue
0.61904762    0.06666667  0.38435439    0.81892837    0.71428571    0.88431513    1.00000000

```

Figure 10. Confusion matrix and classification model statistics for the rpart method.

Regression Models

The following metrics are used to evaluate regression models for each machine learning method (**Figure 11**).

- Mean absolute error (MAE): measures the average magnitude of the errors in a set of predictions, without considering their direction; a lower number indicates better performance
- Root mean-squared error (RMSE): the square root of the average of squared differences between prediction and actual observation; a lower number indicates better performance
- Rsquared: measures the goodness-of-fit of the predicted values to actual values, with zero indicating no predictivity and 100 indicating perfect predictivity

Data in the Uterotrophic.LEL assay has a range of 1e-04 to 1000
The range is important in interpreting the RMSE, which is in the same units as Uterotrophic.LEL
The smaller the RMSE, the better the model

```
Performance metrics for the model: rpart
      RMSE      Rsquared      MAE
282.97683761  0.07095559 190.30012719

Performance metrics for the model: knn
      RMSE      Rsquared      MAE
259.22733308  0.03378955 156.65555384

Performance metrics for the model: svmRadial
      RMSE      Rsquared      MAE
259.45411702  0.02437714 140.35438122

Performance metrics for the model: cforest
      RMSE      Rsquared      MAE
261.21260351  0.05427105 171.47858061

Performance metrics for the model: pls
      RMSE      Rsquared      MAE
268.47185004  0.08473939 171.43160996
```

Figure 11. Results for a regression model in the SummaryReport.txt file.

ModelPerformanceOutPdf.pdf

The graph in this file shows the performance of all machine learning methods using the dataset. Two or more machine learning methods must be selected to generate this file. For classification models, the graph shows the ROC, Sens and Spec (**Figure 12**). For regression models, the graph shows the MAE, RMSE and rsquared (**Figure 13**).

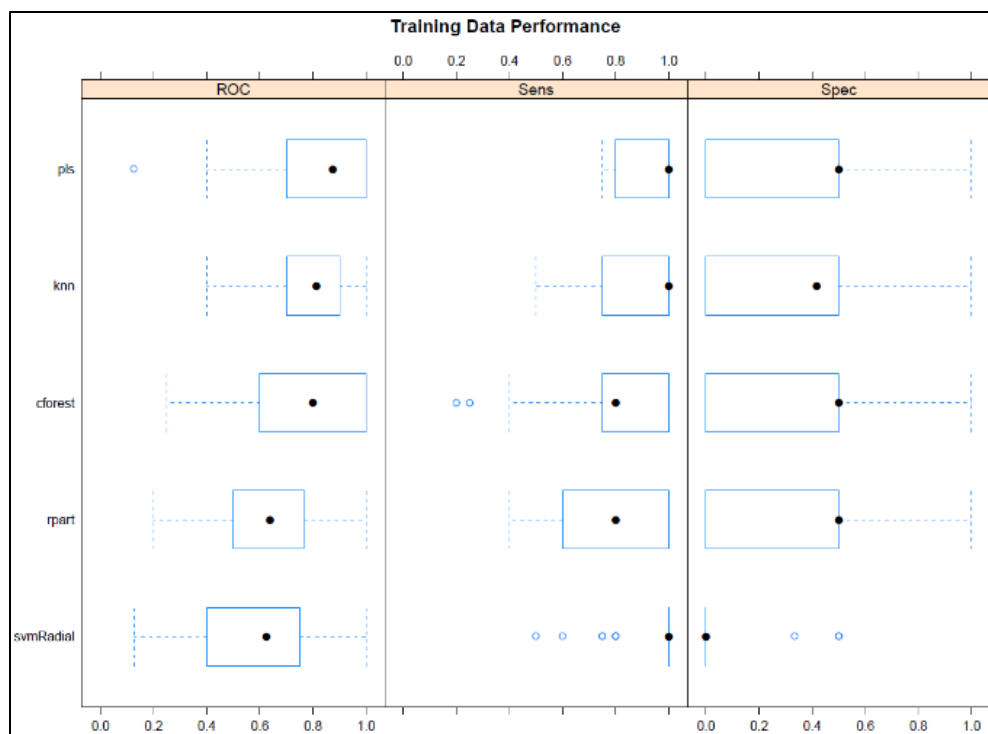


Figure 12. Contents of ModelPerformanceOutPdf.pdf file for a classification model.

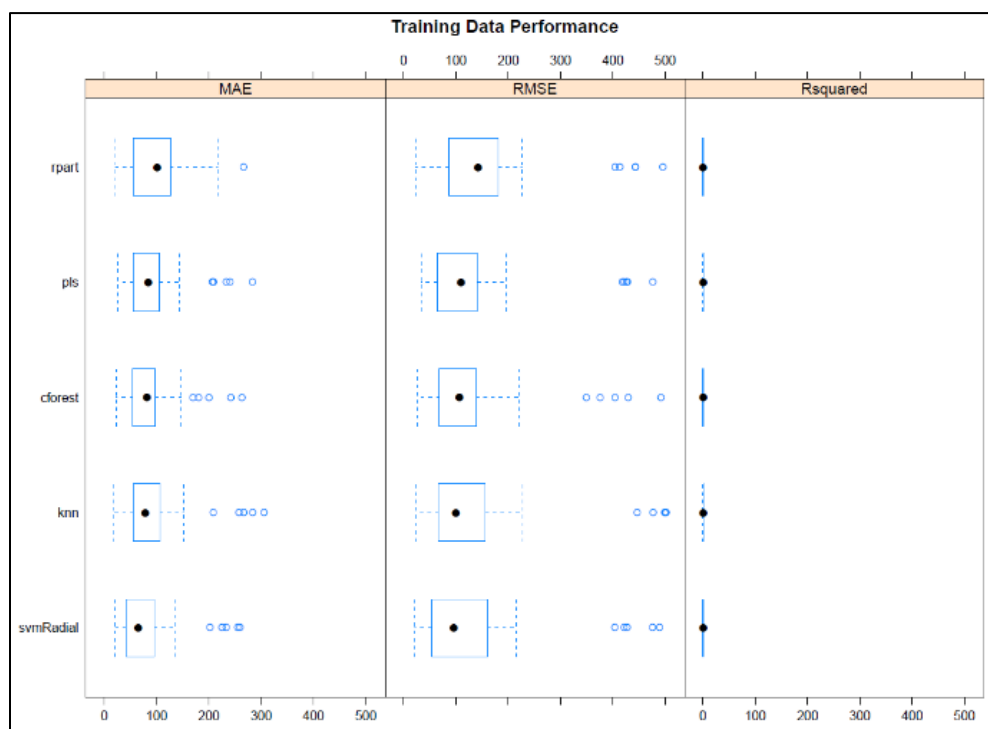


Figure 13. Contents of ModelPerformanceOutPdf.pdf for a regression model

Model Prediction Results

The file PredictionsWithTestDataOut.txt contains the predictions made by each of your machine learning models (**Figures 14 and 15**).

- The first column lists the CASRN for each chemical in the prediction set.
- The second column states whether the chemical is from the user-provided chemical list. Chemicals with a “FALSE” value in this column are the seed chemicals used in model development.
- The next few columns contain the predictions made by each of the machine learning methods you included in your model.
- The "PredictionAssay.Endpoint_groundTruth" column (yellow shading) contains the actual result for each chemical, taken from ICE data. This is what the models are trying to predict. Note that all chemicals with values in this column were used in model development.
- The remaining columns provide data used by the machine learning methods.

CASRN	User provided chemicals	rpart	knn	svmRadial	pls	cforest	LLNA.Call_groundTruth	hCLAT.Call	hCLAT.CD54..EC200	hCLAT.CD54..Call	hCLAT.CD86..EC150
100-06-1	FALSE	Active	Active	Active	Inactive	Inactive	Inactive	Inactive	42.10830369	Inactive	70
100-11-8	FALSE	Active	Active	Active	Active	Active	Active	Active	0.91	Active	0.95
100-39-0	FALSE	Active	Active	Active	Active	Active	Active	Active	2.86	Active	3.2
100-52-7	FALSE	Inactive	Active	Active	Active	Inactive	Inactive	Active	259.9042811	Active	300.2551386
101-86-0	TRUE	Active	Active	Active	Active	Active	Inactive	Inactive	23.35	Inactive	20.3
103-11-7	FALSE	Active	Active	Active	Active	Active	Active	Active	103.77	Active	99.96

Figure 14. PredictionsWithTestDataOut.txt output for a classification model. Model performance statistics reported in other files are based on the concordance of the model predictions with the “LLNA.call_groundTruth” value (yellow shading).

CASRN	User provided chemicals	rpart	knn	svmRadial	cforest	pls	Uterotrophic.LOEL_groundTruth	PhysChem.Properties.MW..Molecular.Weight	PhysChem.Properties.BP
104-40-5	TRUE	25.62445163	167.102631	112.9631975	128.2335052	146.4560042	133	220.1827154	269.1670897
104-43-8	TRUE	25.62445163	65.48509607	57.9214173	85.79582442	68.34257032	39.90699776	262.2296656	277.3367185
10540-29-1	TRUE	25.62445163	8.680774019	3.086136863	33.55915348	-15.77977806	0.1230335	371.2249145	419.3410637
118-58-1	FALSE	25.62445163	159.1559502	108.4587367	99.25323207	147.2009214	3.7	228.0786442	325.5014535
119-61-9	FALSE	237.2943189	202.836388	202.7576907	214.8486841	231.3331792	500	182.0731649	306.490603

Figure 15. PredictionsWithTestDataOut.txt output for a regression model. Model performance statistics reported in other files are based on the concordance of the model predictions with the “Uterotrophic.LEL_groundTruth” value (yellow shading).

Other Output Files

Information in these files can help you better understand why your models performed as they did. These data can also help guide selection of assays and chemicals for another round of model building.

MachineLearning_input.txt

This file lists every data point for each assay/chemical combination you selected as input for your model. “NA” indicates where data were not available for a particular assay/chemical combination.

Variable_Importance.pdf

This file includes one graph for each machine learning method. Each graph (**Figure 16**) shows the relative contribution of each assay to the model performance. The most important variables are at the top of each plot and the least important at the bottom of each plot.

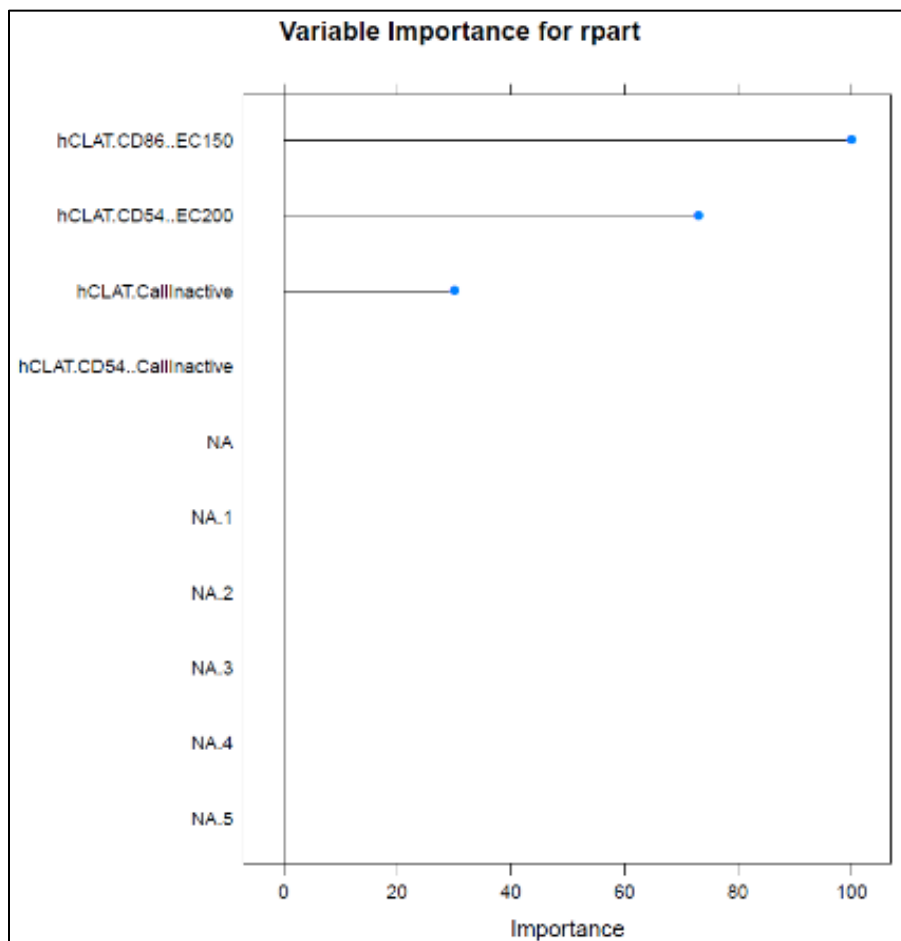


Figure 16. Variable importance for four hCLAT properties for a classification model built using the rpart machine learning method.

userChemicals.txt

This file lists all the chemicals provided as input before processing, including both user-provided chemicals and the seed chemicals provided by ICE.

Appendix 1: How the Machine Learning Tool Works

The ICE machine learning tool is built on the R caret package ([Kuhn 2008](#)).

Before model construction, the machine learning pipeline processes input data as follows:

1. Imputes missing values (replaces them with a statistical estimate)
 - This is done because some machine learning methods will not work with missing data.
 - k-Nearest neighbor (knn) imputation is initially tried, which will work for most ICE datasets.
 - If there is not enough data for knn imputation, median imputation is used.
 - Data removal occurs if after imputation more than 33% of chemicals or assays have no data.
 - At least 10 chemicals and four assay endpoints must exist after data removal to construct a model.
2. Removes highly correlated variables
 - Highly correlated variables may be reporting on the same feature. Allowing these in the model may inflate model performance (cause the model to appear to perform better than it actually does).
 - This step is only performed for numerical data, not categorical data (e.g., TRUE / FALSE).
3. Removes variables with a very restricted data range
 - Variables with near zero variance contribute little to model accuracy and will unnecessarily increase computation time.
 - This step is only performed for numerical data.
4. Performs Z-score standardization
 - Each variable (assay) is transformed to have a mean of 0 and a standard deviation of 1.
5. Creates training and testing datasets
 - Data is randomly split so that 75% of the data is used to build the model (training dataset) and 25% is used to evaluate performance (testing dataset).

The training data set is used to build a model using each machine learning method. Performance is assessed using 10-fold cross validation repeated five times.

- The 10 most important assays are identified as shown in the Variable Importance plot (**Figure 16**).

- Model performance is evaluated and reported as described above in the ModelPerformanceOutPdf.pdf file (**Figures 12 and 13**).

Each machine learning method is then validated using the testing data set. Model performance is evaluated and reported as described above in the SummaryReport.txt file (**Figures 8-11**).

Appendix 2: Machine Learning Algorithms Used By ICE

Machine learning algorithms have several parameters that can be varied to obtain the best performance; this is called model tuning. The theoretical basis and tuning parameters for each algorithm used in the ICE machine learning tool are discussed below.

Recursive Partitioning and Regression Trees

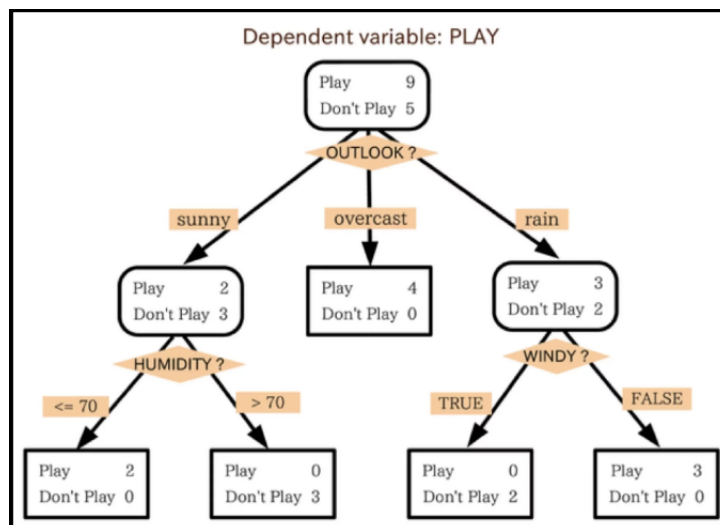
Recursive partitioning and regression trees (rpart) is a decision tree algorithm. A simple illustration of rpart application is provided in **Figure 17** ([Sanjeevi 2017](#), [Benyamin 2012](#)). Data in panel (a) is used as input to determine if we should go outside and play. Input parameters include Outlook (sunny, overcast, or rainy), Temperature, Humidity, and Wind.

A root node is determined and a decision tree is computed as shown in panel (b). An a priori decision is made that we will always go outside and play when it is overcast, so there is no further branching is done when it the Outlook is “overcast”.

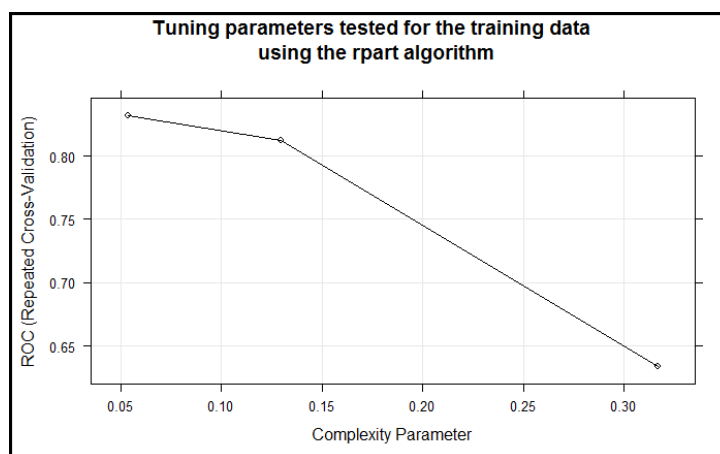
For cases in which the Outlook is either “sunny” or “rainy”, the [R package](#) uses a complexity parameter for tree construction as shown in panel (c). The complexity parameter is used to control the size of the decision tree and to select the optimal tree size. In this example, since the receiver operating characteristic is highest for a complexity parameter of 0.05, this would be used in assessing the performance of the different machine learning algorithms.

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

(a)



(b)



(c)

Figure 17. Data, decision tree and complexity parameter for the rpart example. Panel (a) is the data, panel (b) is the decision tree based on the weather data and (c) is an example of tuning the complexity parameter testing that occurs with the R caret package.

k-Nearest Neighbors

k-Nearest neighbors (knn) is one of the simplest machine learning algorithms. This algorithm makes predictions based on the characteristics of the closest neighbors in the training dataset. Two parameters must be specified:

- How to measure distance (Euclidean distance, cosine similarity, Manhattan distance, etc.)
- Number of nearest neighbors to use

In the example shown in **Figure 18** ([Tolpygo 2017](#)), there are two classes and eight datapoints in the training data. The task is to assign a new datapoint (green circle) to Class 1 or Class 2. If we specify that only one nearest neighbor is to be used to classify the new datapoint, it will be classified as Class 1. However, if we use the three nearest neighbors, the new datapoint will be classified as Class 2 since the nearest neighbors are two Class 2 datapoints and one Class 1 datapoint.

Importantly, knn is a non-parametric algorithm, meaning that no assumptions are made regarding the distribution of underlying dataset. This is in contrast to parametric algorithms where a common assumption is that the data is normally distributed.

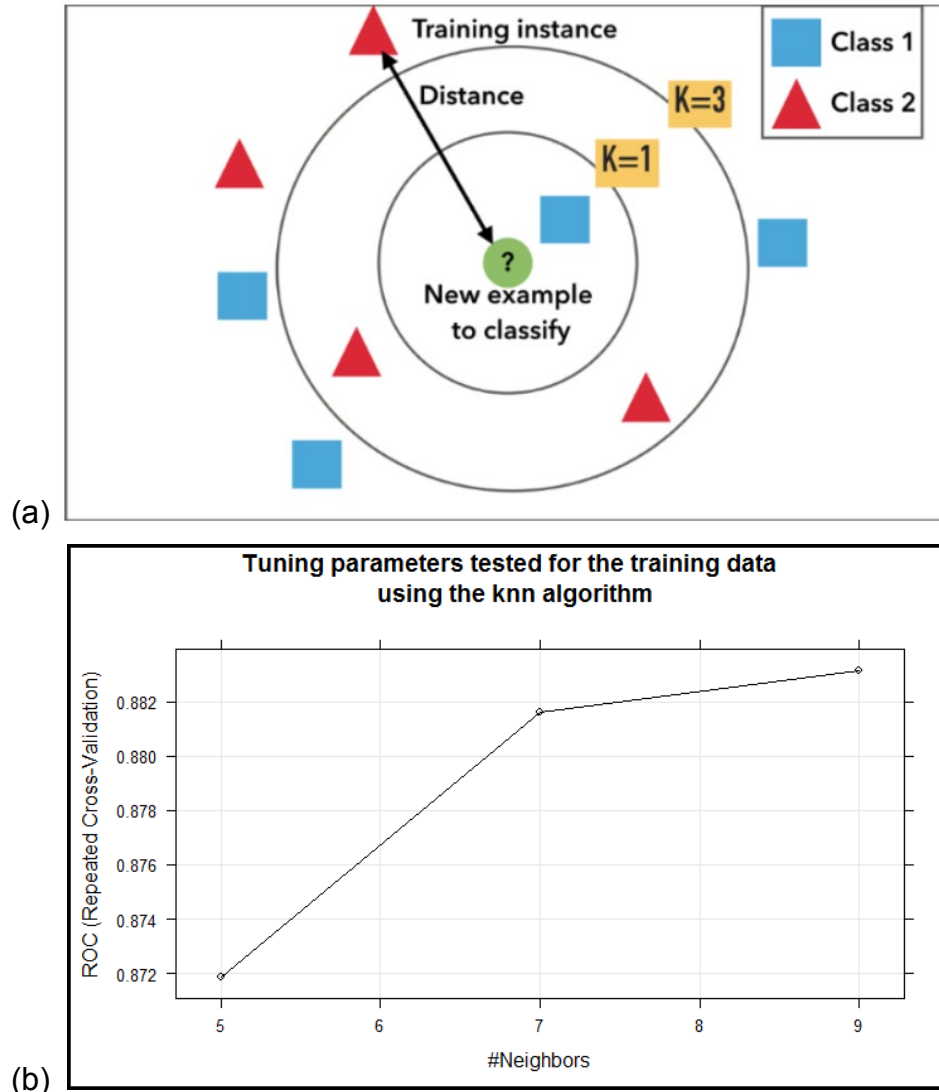


Figure 18. knn example. (a) Two classes of known datapoints and a new datapoint to classify. (b) Tuning parameters used by the caret R package (Tolpygo 2017).

The kkn [R package](#) used in the ICE machine learning tool automatically selects three values for nearest neighbor analysis using the training dataset as shown in **Figure 18**. The numbers of neighbors that yields the best performance is used for the analysis of the testing dataset.

Support Vector Machine with a Radial Kernel

Support vector machine (svm) uses a mathematical method to describe the best separator between two sets. The separator is called a “hyperplane” since data may exist in more than three dimensions but is depicted graphically as a line. svm creates the

best hyperplane by defining a “margin”, the maximum distance from the classification boundary (hyperplane) and the closest training data point.

The example in **Figure 19** panel (a) ([Darkos 2018](#)) shows that there are many hyperplanes that can perfectly separate the two classes. The hyperplane defined by the maximum margin is shown in panel (b), which shows the datapoints closest to the hyperplane are the solid squares and circle.

In ICE, a svm algorithm with a radial kernel is used (svmRadial). This algorithm is available in the caret [R package](#) under kernlab library. The radial kernel can often provide better separation than a lower dimension hyperplane due to how it calculates distance between points. The tuning parameter Cost examines the tradeoff between correct classification and maximization the margin (distance between groups). In panel (c), the Cost value that maximizes the ROC (a measure of the true positive rate vs the false positive rate) is 1 so that would be the value used in modeling.

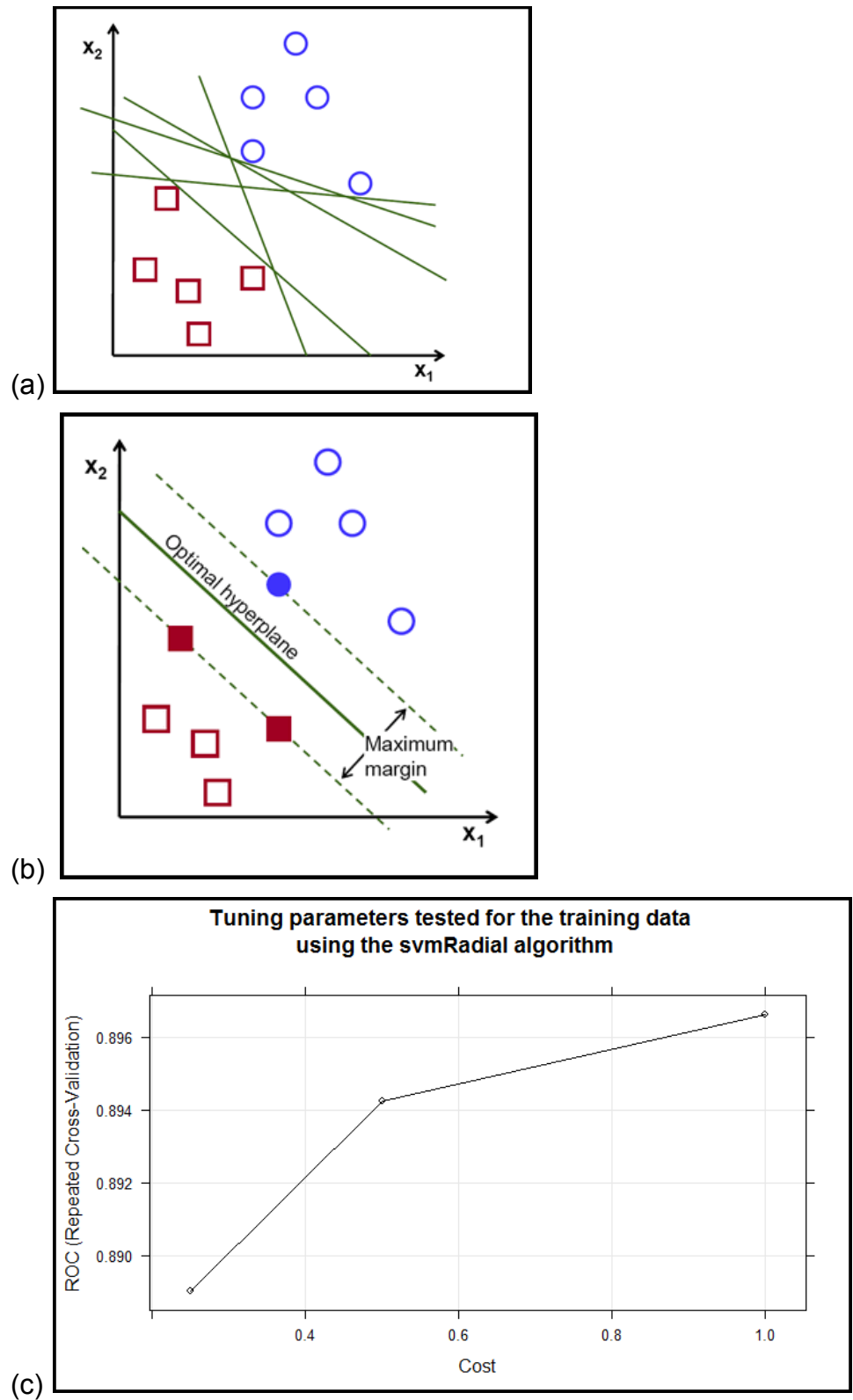


Figure 19. SVM example. (a) Many hyperplanes can be defined to separate the two classes (circles and squares), (b) Hyperplane achieving the optimal separation between the classes (Darkos 2018). (c) Tuning parameters used by caret for the SVM algorithm.

Conditional Random Forest

Random forest (rf) algorithms aggregate multiple outputs from a series of decision trees made by a diverse set of predictors and combine them to make a prediction (Figure 20 panel (a)). ICE uses the conditional rf variant from [R package](#) party, which, unlike RPART, uses a statistical significance test to select variables at each split. The tuning parameter used is the number of predictors from the dataset, with the maximum being the number of datapoints available. A plot of the number of predictors vs. the ROC is shown in panel (b) ([Reinstein 2017](#)). The ROC measures the ratio of true positives vs. the false positives that result from different values of the tuning parameter. In this example, the highest ROC is obtained using 25 predictors.

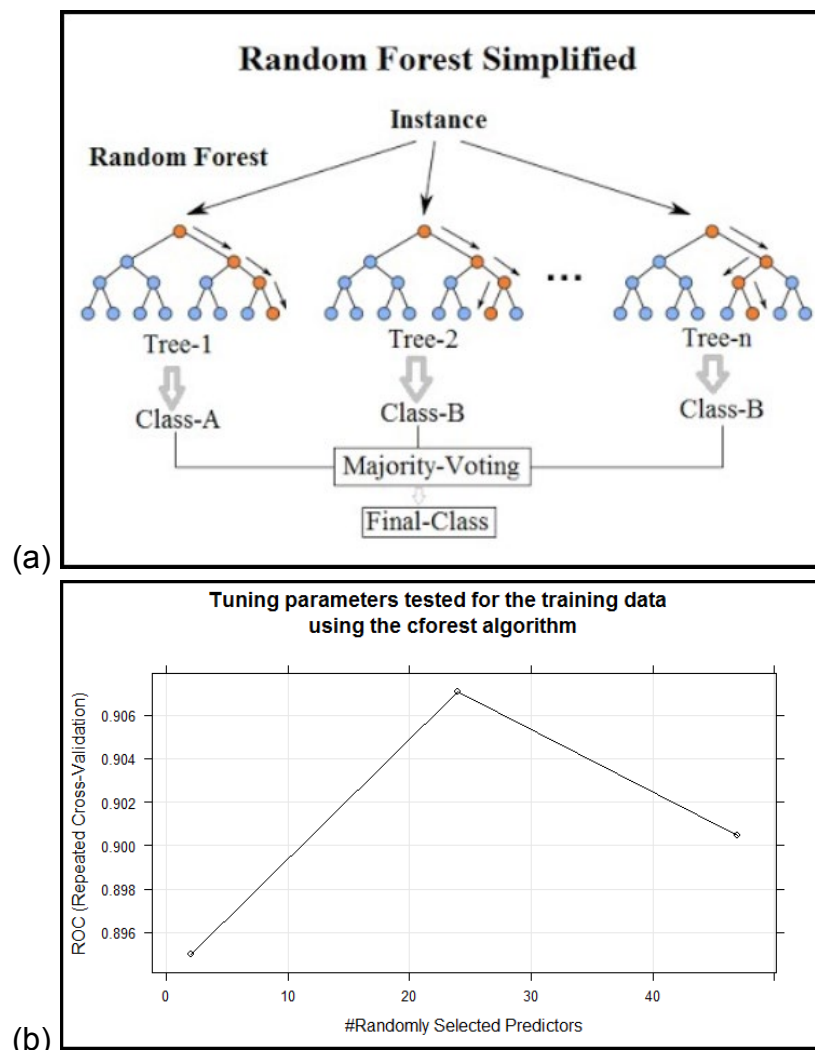


Figure 20. (a) Random Forest uses a series of decision trees. (b) Tuning parameters used by the caret package (Reinstein 2017).

Partial Least Squares

Partial least squares (pls) is a regression technique based on covariance. It is especially useful in cases where there are more variables (assays) than observations (chemicals) and when variables are highly colinear. ICE uses [R package](#) pls, which reduces the variables to a smaller set of uncorrelated components (also called latent variables) and performs least squares regression on these components, instead of on the original data. The tuning parameter specifies the number of components.

Appendix 3: References and Other Resources

References Cited in this Document

Kuhn, M. Building Predictive Models in R Using the caret Package. Journal of Statistical Software, 28(5), 1 - 26, 2008. <http://dx.doi.org/10.18637/jss.v028.i05>

Sanjeevi M. 2017. Chapter 4: decision trees algorithm. In: Deep Math Machine Learning (Sanjeevi M). Medium.com. <https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1>

Benyamin D. 2012. A Gentle Introduction to Random Forests, Ensembles, and Performance Metrics in a Commercial System (Benyamin D.) CitizenNet Blog. <http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics>

Tolpygo A. 2017. Time-Series Analysis: Wearable Devices using DTW and kNN (Tolpygo A.) SFLScientific.com. <https://sflscientific.com/data-science-blog/2016/6/4/time-series-analysis-fitbit-using-dtw-and-knn>

Darkos G. 2018. Support Vector Machine vs Logistic Regression (Darkos G.) towardsdatascience.com. <https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f>

Reinstein I. 2017. Random Forest (Reinstein I.) kdnuggets.com. <https://www.kdnuggets.com/2017/10/random-forests-explained.html>

For More Information

https://en.wikipedia.org/wiki/Machine_learning

<https://towardsdatascience.com/introduction-to-machine-learning-db7c668822c4>

<https://www.coursera.org/learn/machine-learning>

<https://medium.freecodecamp.org/every-single-machine-learning-course-on-the-internet-ranked-by-your-reviews-3c4a7b8026c0>

<http://topepo.github.io/caret/index.html>

<https://www.quora.com/What-is-cross-validation-in-machine-learning>

<https://machinelearningmastery.com/k-fold-cross-validation/>

<https://www.kdnuggets.com/2017/10/random-forests-explained.html>

Resources to Help with Interpreting Results

<https://www.medcalc.org/manual/roc-curves.php>

https://en.wikipedia.org/wiki/Confusion_matrix

<https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>

<http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit>

Appendix 4: Abbreviations

This lists includes both abbreviations used within this User Guide and abbreviations used in the ICE machine learning tool interface.

AC50: concentration at which 50% maximal activity is observed

ACC: activity concentration at cutoff; concentration at which a concentration-response curve exceeds the activity cutoff threshold

AOH: rate constant for the atmospheric, gas-phase reaction between photochemically produced hydroxyl radicals and organic chemicals (units of log₁₀ cm³/molecule-sec)

ARE I_{max} (KeratiNoSens): maximal fold gene induction of the antioxidant response element

ARE EC 1.5 (KeratiNoSens): gene induction of the antioxidant response element representing 50% enhanced gene activity

BCF: fish bioconcentration factor

BP: boiling point

CASRN: Chemical Abstracts Service Registry Number

CD54 (hCLAT): measure of expression of the CD54 cell surface marker

CD86 (hCLAT): measure of expression of the CD86 cell surface marker

CL_{int}: hepatic (in vitro) intrinsic clearance

CV 75% (hCLAT): chemical concentration that results in 75% cell viability compared to solvent/vehicle control

EC150 (hCLAT): chemical concentration that results in 50% increase in activity over control

EC200 (hCLAT): chemical concentration that results in 50% increase in activity over control

EC3 (LLNA): effective concentration inducing a stimulation index of 3

fu: plasma fraction unbound

hCLAT: human cell line activation test

HL: Henry's Law constant (air/water partition coefficient)

HMT: human maximization test

HR IPT: human repeat insult patch test

IC 150 (KeratinoSens): half-maximal inhibitory concentration

KOA: octanol-air partition coefficient

knn: k-nearest neighbor

LogD: octanol-water distribution constant

LogP:: octanol-water partition coefficient

LEL: lowest effective level

LLNA: murine local lymph node assay

LOEL: lowest observed effect level

MAE: mean absolute error

MP: melting point

MW: molecular weight

NOEL: no observed effect level

pKa: acid dissociation constant

pls: partial least squares

rf: random forest

ROC: receiver operating characteristic

RMSE: root mean-squared error

rpart: recursive partitioning and regression trees

Sens: sensitivity

Spec: specificity

svm: support vector machine

VP: vapor pressure

WS: water solubility