

Introduction to class prediction

Class prediction is the process of building a statistical model that is used to predict the class membership of unknown samples. The class prediction model is based on the prior analysis and interpretation of the abundance profiles of features in samples with known class membership.

This *Workflow Guide Overview* is complementary to the *Class Prediction with Agilent Mass Profiler Professional - Workflow Guide* (5991-1911EN). The chapters of the workflow guide follow the flow illustrated in [Figure 1](#).

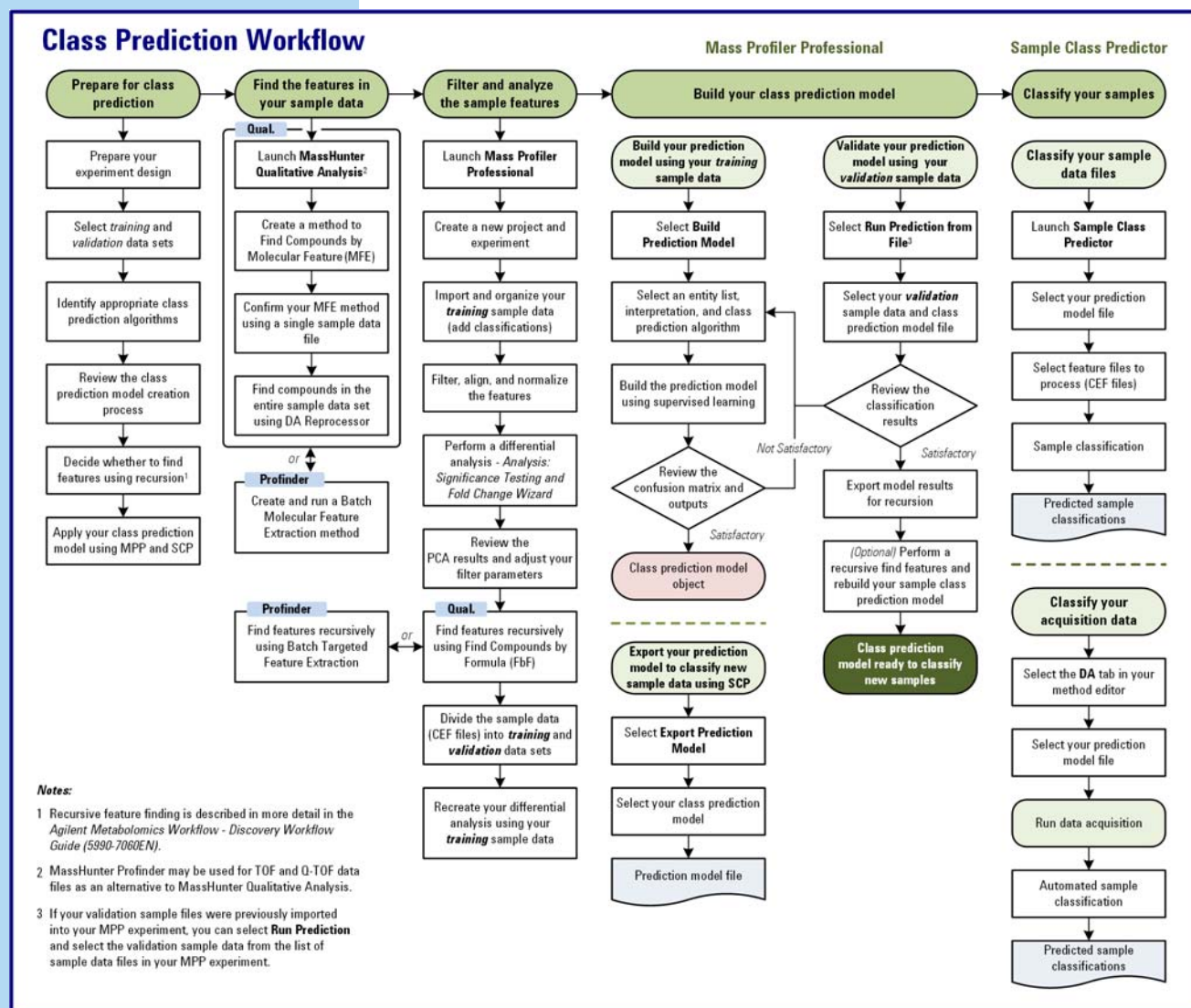


Figure 1 Class prediction workflow

To increase your confidence in obtaining reliable and statistically significant results, review the chapter "Prepare for an experiment" in the *Agilent Metabolomics Workflow - Discovery Workflow Guide* (5990-7067EN) and make sure your analysis includes a carefully thought-out experimental design that includes the collection of replicate samples.

Required items

To build a class prediction model

- Data files from an Agilent GC/MS or LC/MS system
- PC running Windows 7
- Qualitative Analysis/DA Reprocessor or Profinder
- Mass Profiler Professional
- ID Browser

To automate class prediction

- PC running Windows
- MassHunter Data Acquisition or GC/MSD ChemStation Software
- Sample Class Predictor Software

The *Class Prediction with Agilent Mass Profiler Professional* workflow performs best when using the hardware and software described in the “required” sections below. The required hardware and software is used to perform the data acquisition and analysis tasks shown in [Figure 2](#).

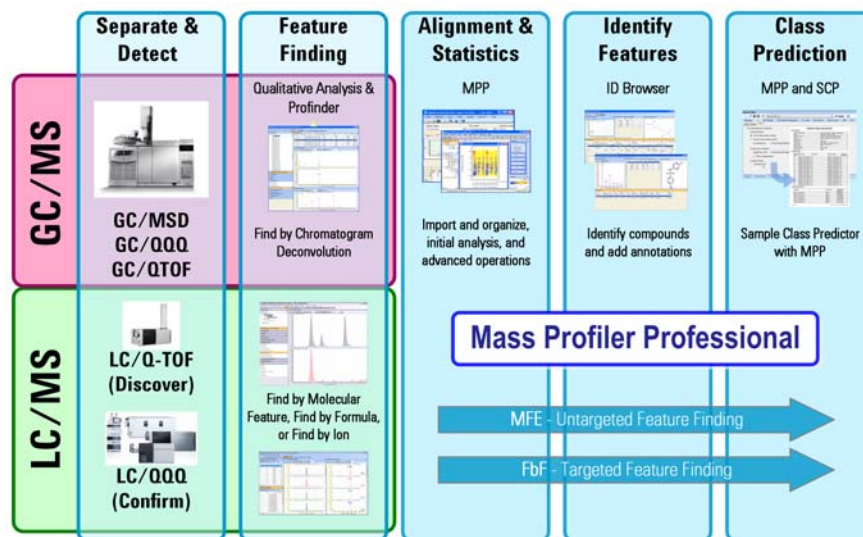


Figure 2 Agilent hardware and software used to acquire and analyze your samples following the class prediction workflow. Sample separation to class prediction typically involves either or both GC/MS and LC/MS analyses.

A typical workflow starts with data acquisition and provides you with the tools through to analysis involving both untargeted (discovery) LC/MS and targeted (confirmation) LC/MS/MS analyses. Molecular feature extraction (MFE) and Find by Formula (FbF) are two different algorithms used by MassHunter Qualitative Analysis and Profinder for finding compounds. All results files generated by Agilent analytical platforms can be imported into Mass Profiler Professional for quality control, statistical analysis and visualization, interpretation, pathway analysis, and class prediction.

Agilent MassHunter Mass Profiler Professional is used in conjunction with Agilent MassHunter Qualitative Analysis/DA Reprocessor or Agilent MassHunter Profinder to build and validate class prediction models. After you create your class prediction model using Mass Profiler Professional, you can use Mass Profiler Professional to manually run class prediction to classify new samples.

Agilent MassHunter Sample Class Predictor adds value by automating your sample class prediction with acquisition and helping you save time and money in your analyses by providing real-time quality assurance and control of your data. With Sample Class Predictor you can integrate the class prediction models you create using Mass Profiler Professional within Agilent Data Acquisition and automatically classify new samples.

Required hardware

- PC running Windows
 - *Minimum:* Windows 7 (32-bit or 64-bit) with 4 GB of RAM
 - *Recommended:* Windows 7 (64-bit) with 8 GB or more of RAM
 - At least 50 GB of free space on the C:\ partition of the hard drive
- An Agilent chromatography mass spectrometry system (for example, GC/MS, LC/MS, and CE/MS) to generate the sample data files used in this workflow.

Required software

- Mass Profiler Professional Software B.12.00 or later
- MassHunter Qualitative Analysis software, Version B.06.00 SP1 or later
- MassHunter Profinder software, Version B.06.00 or later
- MassHunter Data Acquisition software, Version B.06.00 or later (this includes MassHunter DA Reprocessor)
- MassHunter ID Browser B.05.00 or later

Optional software

- MassHunter Quantitative Analysis software, Version B.06.00 or later
- Agilent ChemStation software
- AMDIS
- METLIN Personal Compound Database and Library
- Agilent Fiehn GC/MS Metabolomics Library

What is class prediction?

Class prediction is the process you use within Mass Profiler Professional (MPP) to build, validate, test, and export a class prediction model that is developed based on the abundance profiles of the features in samples with known classification. The class prediction model created within MPP is subsequently used by Sample Class Predictor (SCP) to integrate class prediction within data acquisition. Using this latter, separately licensed program, you automate the prediction model for real-time QA/QC of your samples. If you do not have a Sample Class Predictor license you can manually process all of your data within Mass Profiler Professional.

The class prediction workflow guide helps you use MPP to build a prediction model that is used to classify samples acquired using chromatography/mass spectrometry. The available prediction model algorithms learn from samples that have known functional class membership (training and validation data sets) to build a prediction model that classifies new samples (test data sets) into the known classes. Class prediction involves ten main steps:

- (1) Prepare your experiment design to include a large number of replicates for each of the known classifications so that the sample data cover a range of variables such as operator, instrument condition, run order, sample preparation, and subject,
- (2) Find the molecular features in *all* of your sample data files using Qualitative Analysis/DA Reprocessor or Profinder,
- (3) Create a differential analysis using Mass Profiler Professional,
- (4) Recursively find the targeted features in all of your sample data files using Qualitative Analysis/DA Reprocessor or Profinder,
- (5) Partition your sample data into *training* and *validation* data sets,
- (6) Recreate your differential analysis with the *training* data set using Mass Profiler Professional,
- (7) Select one or more prediction model algorithms that support your hypothesis, experiment design, and expected interrelationships of the features among the classifications,
- (8) Build your class prediction model using the *training* data set and supervised learning,
- (9) Validate your class prediction model using the *validation* data set, known samples that were not used during the model creation, and
- (10) Apply your prediction model to samples with unknown classification.

Class prediction assigns functional classifications to your samples based on the abundance profile of the features (compounds) identified in your training data set. Possible classifications can include material origin, biological material maturity, production quality, clinical and/or sample treatments, diseases, and other conditions. Class prediction helps you:

- Predict the class membership (parameter value) of a sample
- Identify chemical (metabolite) signatures that discriminate well among classes
- Identify samples that could be potential outliers

A simple view of the class prediction workflow is illustrated in [Figure 3](#) on page 6 starting with data acquisition through to class prediction involving either LC/MS or

GC/MS analyses. Find by Molecular Feature, also referred to as molecular feature extraction (MFE) and Find by Formula (FbF) are two different algorithms used by MassHunter Qualitative Analysis for finding compounds. All results files generated by Agilent analytical platforms can be imported into Mass Profiler Professional for quality control, statistical analysis and visualization, and interpretation.

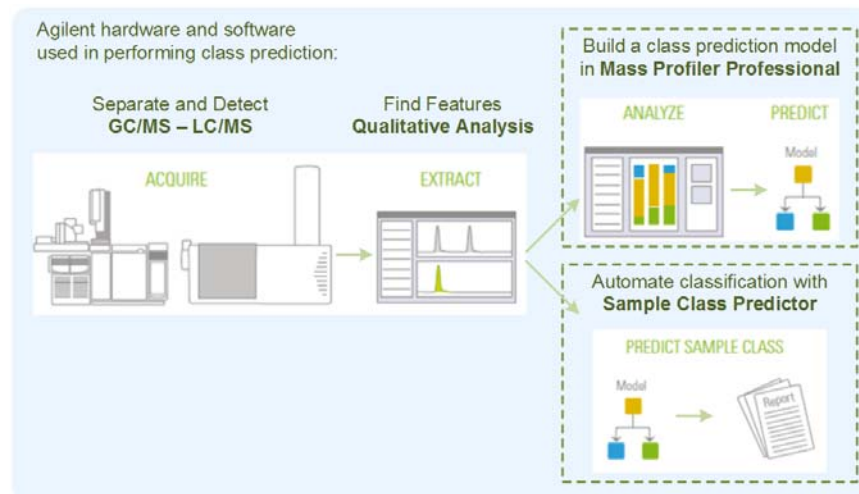


Figure 3 An Agilent class prediction workflow from separation to classification involving either GC/MS or LC/MS analyses.

Experiment variables

Experiment variables and classifications are derived from your experiment. Manipulated attributes of the state of the organism are referred to as **independent** variables. The biological response to the change in the attributes may manifest in a change in the metabolic profile. Each metabolite that undergoes a change in expressed concentration is referred to as a **dependent** variable. Metabolites that do not show any change with respect to the independent variable may be valuable as control or reference signals.

The features in a sample may be individually referred to as a **compound, descriptor, element, entity, feature, or metabolite** during the various steps of the class prediction workflow. When hundreds to thousands of dependent variables (e.g., metabolites) are available, chemometric data analysis is employed to reveal accurate and statistically meaningful correlations between the attributes (independent variables) and the metabolic profile (dependent variables). Meaningful information learned from the metabolite responses can subsequently be used for clinical diagnostics, for understanding the onset and progression of human diseases, and for treatment assessment. Therefore, metabolomic analyses are poised to answer questions related to causality and relationship as applied to chemically complex systems, such as organisms.

Sample sets

Robust prediction models are developed using large sample sets. Each sample set should contain replicate samples from all of the known classifications so that the

Experiment design

sample data cover a range of variables such as operator, instrument condition, run order, sample preparation, and subject. Models developed from samples that cover a sufficiently large range of classifications and contain a large number of replicates can be considered to be generic to the population. The class prediction model developed using these samples can therefore be expected to be able to classify any biological sample taken from the population.

A robust experiment design has the following features.

The hypothesis: The first and most important step in your experiment is to formulate the question of correlation that is answered by the analysis - the hypothesis. This question is a statement that proposes a possible correlation, for example a cause and effect, between a set of independent variables (classifications) and the resulting metabolic profile (compounds, features, entities, or descriptors).

Natural variability: Before you begin collecting your samples it is important to understand how any one sample represents the population as a whole. Because of natural variability and the uncertainties associated with both the measurement and the population, no assurance exists that any single sample from a population represents the mean of the population. Thus, increasing the sample size greatly improves the accuracy of the sample set in describing the characteristics of the population and improves your class prediction model.

Replicate sampling: Sampling the entire population is not typically feasible because of constraints imposed by time, resources, and finances. On the other hand, fewer samples increase the probability of concluding a false positive or false negative correlation. At a minimum, it is recommended that your analysis include ten (10) or more replicate samples for each attribute value for each condition in your study.

System suitability: System suitability involves collecting data to provide you with a means to evaluate and compensate for drift and instrumental variations to assure quality results. The techniques that produce the highest quality results include (1) retention time alignment, (2) intensity normalization, (3) chromatographic deconvolution, and (4) baselining. However, even the best analysis techniques cannot compensate for excessive drift in the acquisition parameters. The best results are achieved by maintaining your instrument and using good chromatography.

Sampling methodology: Improved data quality for your analysis comes from matching the sampling methodology to the experimental design so that replicate data is collected to span the attribute values for each condition. A larger number of samples appropriate to the population under study results in a better answer to the hypothesis and a better class prediction model. An understanding of the methodologies used in sampling and using more than one method of sample collection have a positive impact on the significance of your results.

Class prediction algorithms

Supervised learning

Class prediction algorithms

Class prediction involves a process of assigning a condition (also referred to as an attribute value, parameter value, or class) to a new sample on the basis of a mathematical/statistical model created using a training sample set of data whose conditions are known. Therefore class prediction is an instance of machine learning that is based on supervised learning, i.e., creating mathematical representations of class membership by applying a class prediction algorithm to a set of correctly identified samples. Cluster analysis (referred to as clustering in MPP), on the other hand, is an example of unsupervised learning where the samples are grouped into categories based on some measure of inherent similarity without any prior knowledge of sample identification. To build a class prediction model you must select a class prediction algorithm for the supervised learning that is appropriate to your experiment.

Supervised learning is a process employed in data analysis that uses knowledge of the phenotype to simplify the data (for example, reduce the number of entities) to retain the entities that provide the best correlation to the characteristics (conditions) involved in the particular analysis. The goal of supervised learning is to optimize a mathematical relationship that accurately associates the entities in your samples to the conditions in your interpretation. In order to perform supervised learning your training data must be properly classified.

MPP supports five different class prediction algorithms (machine learning algorithms). The advantages for using any one of the supervised learning algorithms is often a matter of subjective opinion and your personal experience in applying the particular algorithm to your experiment and experimental parameters. The available class prediction algorithms are:

1. Partial Least Squares Discrimination
2. Support Vector Machine
3. Naïve Bayes
4. Decision Tree
5. Neural Network

Except for Partial Least Squares Discrimination, the algorithms used by Class Prediction are also available within the advanced operation Find Minimal Entities.

In general, each of the class prediction algorithms, also referred to as supervised learning algorithms, have the following features. See the section “Identify appropriate class prediction algorithms” in the *Class Prediction with Agilent Mass Profiler Professional - Workflow Guide* ([5991-1911EN](#)) for additional information for each of the algorithms.

Partial Least Squares Discrimination (PLS-DA)

PLS-DA is best suited for making classifications where all of the parameter values are measurable with little error, where the number of samples is smaller than the number of parameter values, and where there may be a simple model structure among the classifications and their attributes. This approach is applicable when an interpretation contains categorical parameter values.

PLS-DA is an extension of partial least squares regression (PLSR). Partial least squares analyses seek to find a linear regression model by projecting the predicted attribute variables and the observed attribute variables onto a new space. PLS-DA is a variant of PLSR where the Y values used in regression are categorical rather than continuous.

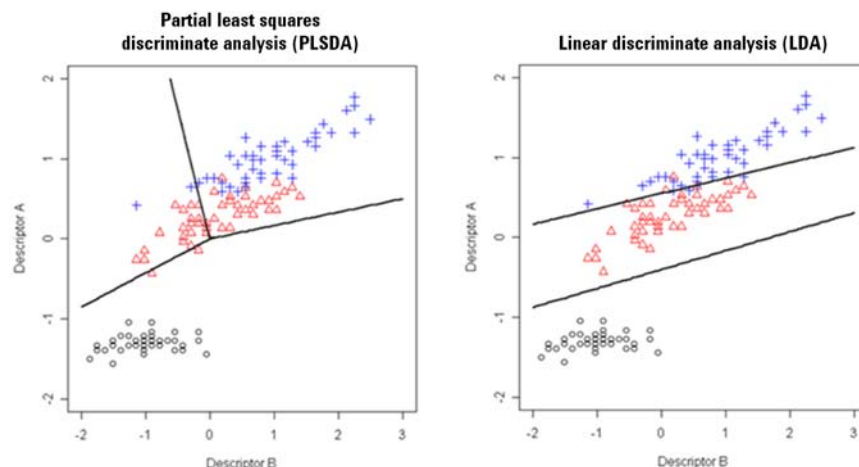


Figure 4 Partial least squares discriminant analysis seeks to find linear combinations of the variables that are highly correlated, but each linear combination does not need to be separated linearly across the entire variable space as in the case of linear discriminate analysis. An entity in this analysis is referred to as a descriptor.

Since PLS-DA is best for making classifications where all of the attribute values are measurable with little error and where there may be a simple model structure among the classifications and their attributes, PLS-DA is well suited for chromatography mass spectrometry. Chromatography mass spectrometry provides an accurate measurement of retention time, mass, and abundance. More-or-less traditional mathematical relationships can be applied to the data, and simple models can be assessed through a least squares regression fit of the data to find the model that provides the best fit.

Support Vector Machine (SVM)

SVM is best suited for making classifications where the attributes within the samples are intertwined and can benefit from transforming the input attribute space into additional dimensionality to identify classification separation planes. The algorithm looks for differentiation among your classifications in pairs and can work with smaller entity lists.

SVM attempts to separate samples representing two classes by imagining each sample as a point positioned in a two or three dimensional space and then calculating the parameters for a line or plane (linear or non-linear) that separates the samples into each classification. While several possible separating planes within a three-dimensional feature space may exist, the SVM algorithm finds the separator that maximizes the separation between the classes encompassing the sample points. The power of SVM stems from the fact that SVM supervised learning can effectively separate samples using non-linear functions and can therefore separate out samples containing intertwined feature sets. SVM therefore can efficiently clas-

sify samples using non-linear classifications by mapping the input into high-dimensional feature spaces.

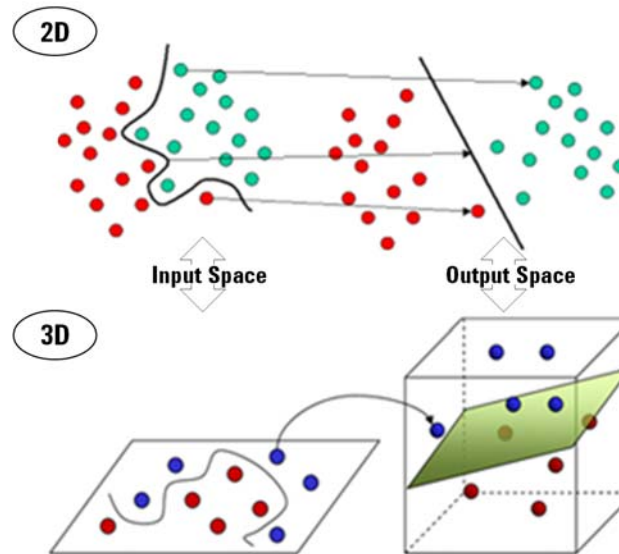


Figure 5 Support vector machine separates samples representing two classifications within a feature space of two and three dimensions.

Naïve Bayesian (NB)

NB is best suited for making classifications where the attributes within a sample are independent from each other. The algorithm looks for differentiation among entities in your entity list by assuming that the change in the appearance of any one entity is unrelated to other entities in the entity list. This algorithm can work with smaller entity lists.

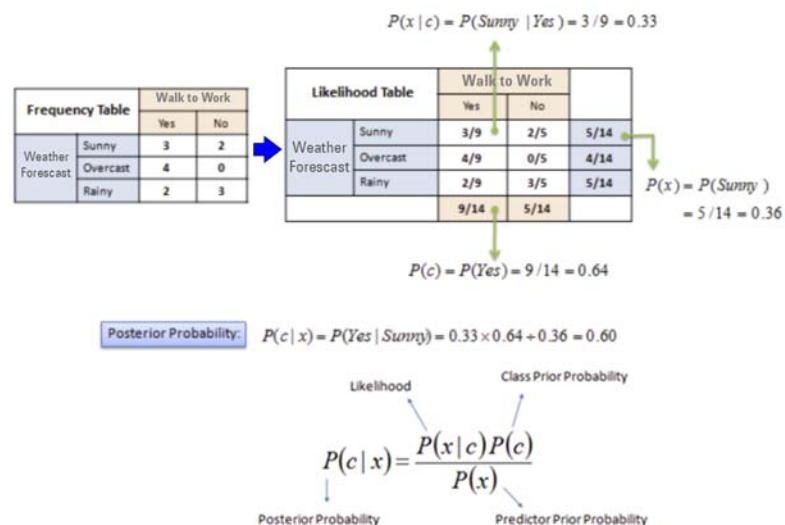


Figure 6 Naïve Bayesian calculation of the probability of walking to work.

Decision Tree (DT)

DT is best suited for making classifications when the attributes are present in all or nearly all of the samples. Classification proceeds through a linear progression of if-then-otherwise decisions based on the attribute values of the samples. An example is illustrated by the following sequence of decisions based on the data in [Table 1](#).

If Feature 1 is greater than or equal to 4, then the sample is part of Class A.
Otherwise, if Feature 1 is less than 4, and

If Feature 2 is greater than or equal to 10, then the sample is part of Class B.
Otherwise, if feature 2 is less than 10, then the sample is part of Class C.

Table 1 Example data set

	Feature 1	Feature 2	Feature 3	Class Label
Sample 1	4	6	7	A
Sample 2	0	12	9	B
Sample 3	0	5	7	C

The sequence of if-then-otherwise decisions can be arranged as a tree called a decision tree. Only two of the three available features in [Table 1](#) were necessary to separate the three classes.

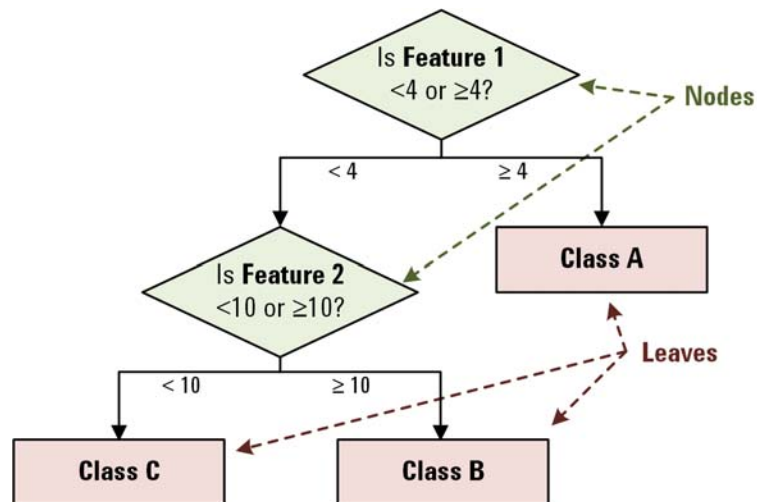


Figure 7 Features of a Decision Tree: nodes are where a decision is made regarding the attributes of a feature, a branch is the path from one node to another node or one node to a leaf, and a leaf is a classification at the end of a branch. This decision tree is created from the data in [Table 1](#).

Neural network (NN)

NN is best suited for making classifications when there likely exists a complex and unknown, or intermediate, layer of relationships that link that attributes within a sample to the classification. The intermediate layer of relationships behaves in a manner similar to how neurons create neural networks in a central nervous system - applying adaptive weights to the numerical parameters and non-linear approximations to the inputs. This algorithm can only be used with interpretations that contain

numerical parameter values (parameter values that are not categorical) and requires a lot of computation time. NN is suitable when the data set involves many sample classifications.

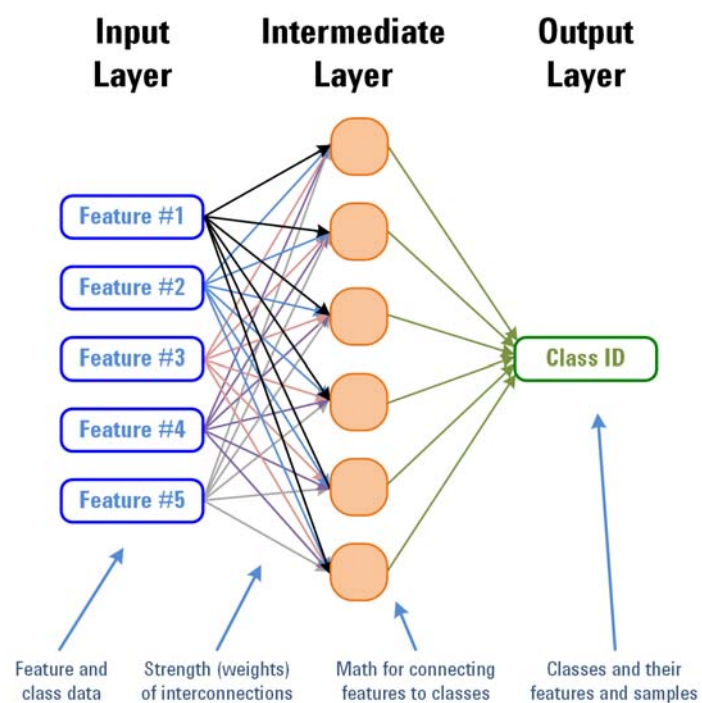


Figure 8 Simple illustration of the layers of a neural network model.

Building your class prediction model

MPP guides you through a sequence of five steps to build your class prediction model (Figure 9).

1. Enter input parameters.
2. Enter validation parameters.
3. Review validation algorithm outputs.
4. Review training algorithm outputs.
5. Review the class prediction model.

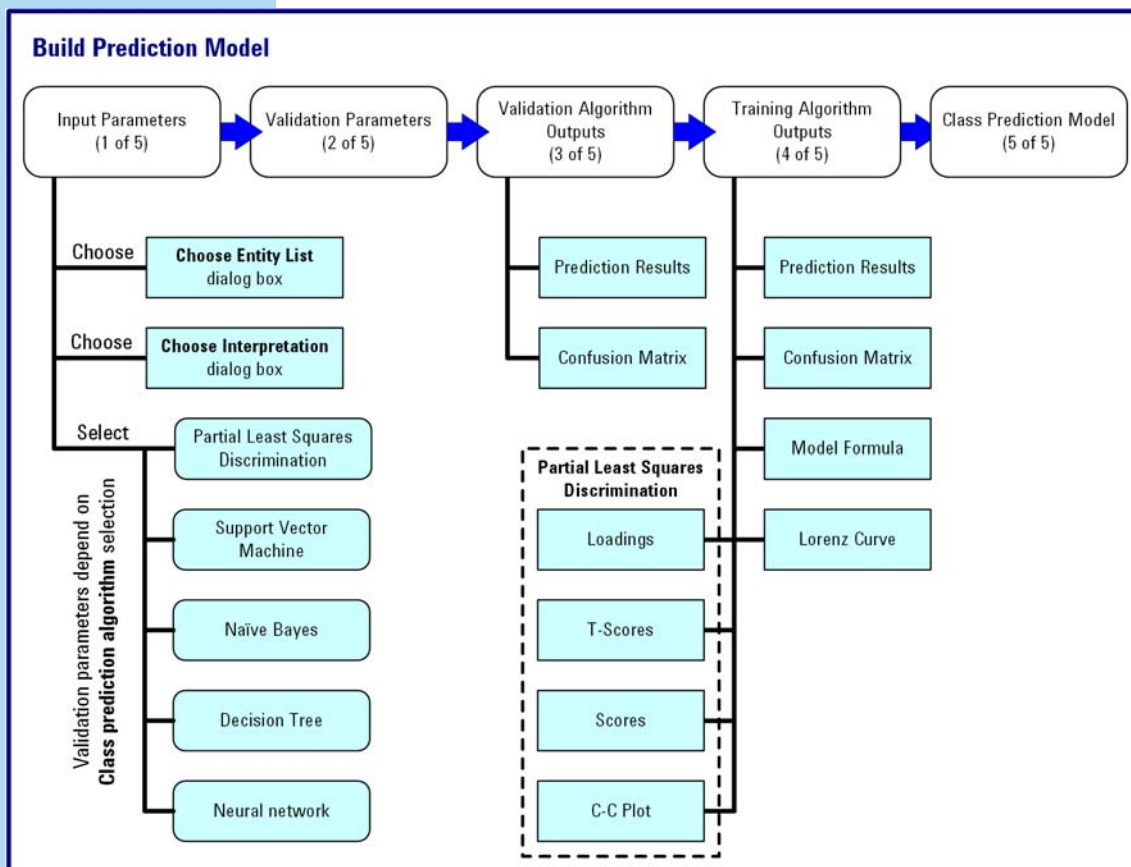


Figure 9 Flow chart of the Build Prediction Model wizard

1. Enter input parameters.

Build your prediction model by first selecting an interpretation, an entity list, and a class prediction algorithm.

The advantages for using any one of the class prediction algorithms is often a matter of subjective opinion and your personal experience in applying the particular algorithm to your experimental parameters.

To build a class prediction model you should use a reliable, large entity list. A reliable entity list to use for class prediction is an entity list that has been filtered to contain entities that appear in no less than one half of the training sample data files, and preferably the entities in your input entity list appear in all of the training sample data files.

2. Enter validation parameters.

Supervised learning validation types

Select the class prediction algorithm parameters and select a supervised learning validation type.

Supervised learning is important to assess the ability of the prediction model to properly classify your samples, but it is also an important tool to avoid underfitting or overfitting your model on the training data as illustrated in Figure 10. A model that is improperly fit typically produces low accuracy with new data because the model fails to fit the underlying functions that classify the data.

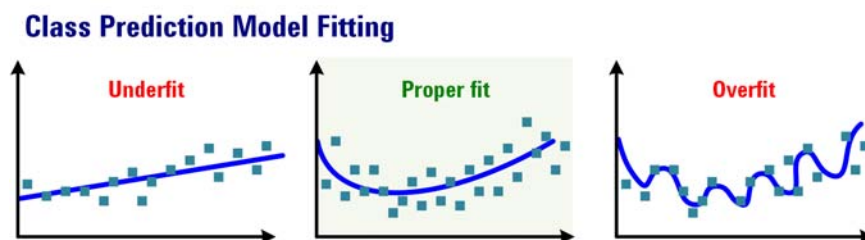


Figure 10 Examples of model fitting in class prediction

Supervised learning is performed using the sample data set that you use to train your model; your training sample data is divided into a uniform set of data whereby each set contains a sample representing each classification and each of these sets in turn is treated as a validation data set to a model that is generated using the remaining sample data sets. The results of the supervised learning are presented in a confusion matrix.

In MPP the prediction accuracy of your class prediction models are evaluated using one of two **validation types**. The validation types organize your training data files into smaller groups and then in turn use the smaller groups within the model development to train and validate the model many times to produce statistical prediction accuracy metrics. This validation approach is most successful when your experiment design includes many replicates for each parameter value. Each validation type presents the performance of the prediction model using a confusion matrix. The available validation type during your prediction model development may vary based on the selected prediction algorithm.

Leave One Out: The replicate samples in your training data set are arranged in columns representing each of the known classifications. All of the data, with the exception of one row containing one sample from each classification, is used to train the learning algorithm and generate a prediction model. The prediction model is then used to classify the remaining row of data as a temporary validation data set. The process is repeated for every row in the dataset and a confusion matrix is generated. During the Leave One Out validation each row of data is sequentially treated as a temporary validation data set for a corresponding model that is trained using all of the other rows of data. Leave One Out validation does not require any parameters necessary to enter or adjust.

N-Fold: The total number of samples associated with the training data set are randomly divided into N equal parts. The samples representing N-1 parts are used for training your prediction model and the remaining samples, representing the remaining one part, are used for validating the prediction model. The process is repeated N times, with a different part of the training data set being used to validate the model

3. Review validation algorithm outputs.

developed using the samples representing the N-1 parts. Using N-fold, like Leave One Out, each sample is used at least once to validate the prediction model. A confusion matrix is generated from the results. The default values of three-fold validation and one repeat are sufficient for most analyses. For results with greater confidence, you can use a ten-fold validation with three repeats. For large data sets, the latter may take significantly more computing time.

Review the class prediction results and the confusion matrix generated from the internal validation. Remember that this validation is performed by segregating your training data sets into smaller groups.

The prediction results report provides details of the prediction versus actual classification for each condition. The Confusion Matrix results show a cumulative Confusion Matrix, which is the sum of confusion matrices for individual runs of the learning algorithm and a summary of the expected efficacy of the prediction model. After you review the confusion matrix to assess the accuracy of the prediction model, decide whether to make further changes to the parameters and repeat the supervised learning.

4. Review training algorithm outputs.

The performance results include prediction results, a confusion matrix for the training model on the whole entity list, and a Lorenz curve showing the efficacy of classification and the prediction model. The Confusion Matrix results in this step show the result of applying the final prediction model to the training sample data. Wherever appropriate, a visual output of the classification model is presented.

5. Review the Class Prediction Model.

Review the summary information for your class prediction model and save the class prediction model.

Recursive feature finding

During the class prediction workflow you recursively find the features in your original sample data files. Combined with collecting replicate samples in your experiment, recursive feature finding improves the statistical accuracy (confidence) of your analysis and reduces the potential for obtaining a false positive or a false negative answer to your hypothesis and sample classification.

The following steps are typical for differential analysis and class prediction and are presented to help you understand where recursion is performed.

- a **Begin an MPP-based workflow.** MPP experiments typically begin with analyzing untargeted features that were subsequently binned together during the alignment in the MS Experiment Creation Wizard.
- b **Perform an initial differential analysis.** The goal of the initial differential analysis is to find a correlation of the intensity variations among the untargeted features to the sample conditions (classifications). The large number of untargeted features found in the sample data files is reduced to a much smaller number of features that are now considered more important to further analyses.
- c **Perform recursive feature finding.** Recursive feature finding improves the confidence measure of your analysis. To perform recursive feature finding you export

the features from the initial differential analysis as a targeted list of features. This step improves the quality of finding the features in the original sample files; targeted feature finding focuses on finding a specific set of features with less emphasis on feature strength.

Performing recursion is always recommended, and is especially recommended when your class prediction model depends on features that show up and down regulation among your classification or the presence and absence of features.

- If your prediction model relies on the regulation of strong features, enhanced finding of weak targeted features may not significantly help you during your initial model creation and recursive feature finding may be optional before creating your initial class prediction model.
- If your prediction model relies on feature presence and absence then it is recommended you perform recursive feature finding before creating your initial class prediction model.
- If your hypothesis does not include a priori knowledge on the regulation or absence of features in your samples then it is recommended you perform recursive feature finding before creating your initial class prediction model.

d **Recreate your initial class prediction model.** During this step you gain additional confidence in your analysis through the use of targeted features that were identified as significant during your initial differential analysis.

e **Build your class prediction model.**

f **(Optional) Perform recursive feature finding.** If you did not perform a recursive feature finding after your initial differential analysis, export the prediction model features from your class prediction model for recursive feature finding. Class prediction confidence benefits from targeted feature finding to find the features in your sample files.

- Export the prediction model features from your class prediction model.
- Recursively find the features used by your class prediction model in your sample data files.
- Recreate your experiment, feature filters, differential analysis, and class prediction model.

g **Apply your class prediction model.**

Apply your class prediction model

After you have created and saved your class prediction model, you can run your prediction model on your validation data set and new sample data. When you run your prediction model on your validation data set (samples that were set aside during your model creation that have known class membership but were not part of the training data set), you are performing a final quality inspection of your class prediction model.

Three ways are available to classify new sample data files that have previously been acquired:

1. **MPP - Run Prediction:** Classify samples that are part of an experiment but were not part of building or validating the prediction model.
2. **MPP - Run Prediction from File:** Classify samples that reside in a folder on your computer or on a network drive.
3. **SCP - Project > Run Prediction:** Classify samples that reside in a folder on your computer or on a network drive using the same **Run Prediction from File** wizard used by MPP. SCP is a separately licensed program.

Layout of the Mass Profiler Professional screen

The main functional areas of the Mass Profiler Professional screen are illustrated in [Figure 11](#) on page 18. Your class prediction model appears in the experiment navigator within the display pane.

The main Mass Profiler Professional window consists of four parts:

Menu Bar - access to actions that are used for managing your projects, experiments, pathways, and display pane views

Toolbar - access to buttons for commonly used tasks grouped by project, experiment, entity, statistical plot, and sidebar tasks

Display Pane - organized into functional areas that help you navigate through your project, experiments, analyses, and available operations

Status Bar - information related to the current view, cursor position, entity, and system memory

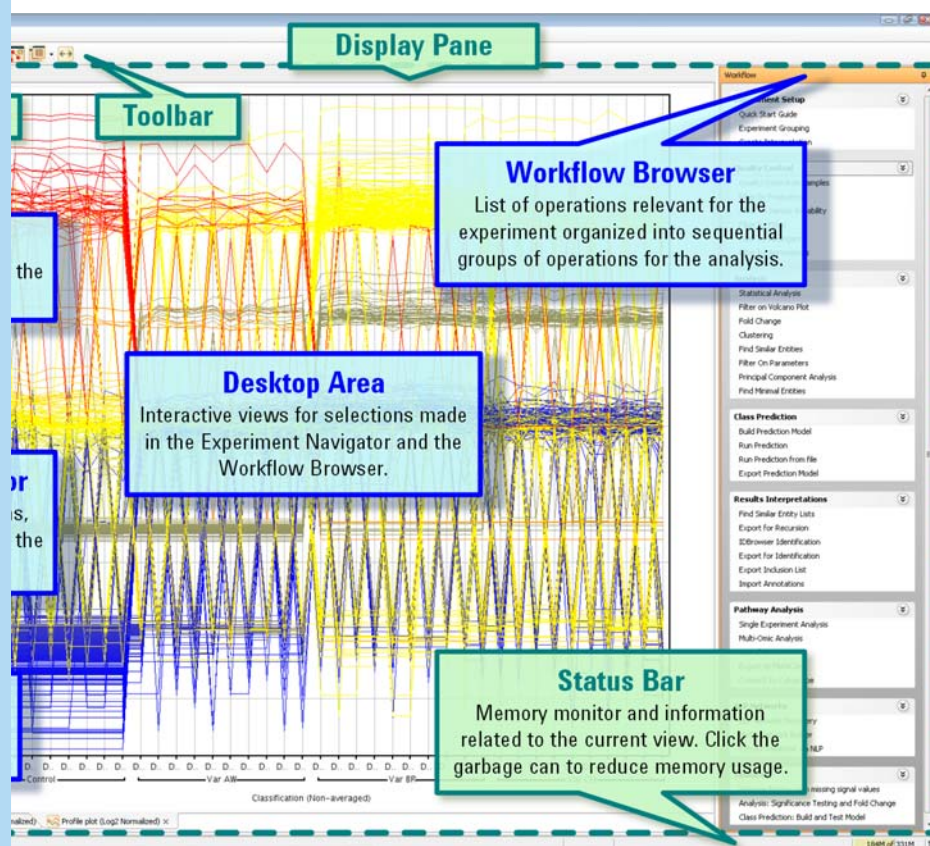


Figure 11 The main functional areas of Mass Profiler Professional

For more information

The *Class Prediction with Agilent Mass Profiler Professional - Workflow Guide* is part of the collection of Agilent manuals, help, application notes, and videos. The current collection of manuals and help are valuable to users who understand class prediction and the metabolomics workflow and who may require familiarization with the Agilent software tools. Video tutorials for MPP provide step-by-step instructions to analyze example GC/MS and LC/MS data files. This workflow provides a step-by-step overview of performing class prediction using MassHunter Qualitative Analysis, Profinder, and Mass Profiler Professional.

The following selection of publications provides materials related to class prediction, metabolomics, and MassHunter software used to analyze your sample data:

- *Manual: Agilent G3835AA MassHunter Profinder Software - Quick Start Guide* ([G3835-90014](#), Revision A, December 2013)
- *Manual: Integrated Biology with Agilent Mass Profiler Professional - Workflow Guide* ([5991-1909EN](#), Revision A, June 2013)
- *Manual: Integrated Biology with Agilent Mass Profiler Professional - Workflow Guide Overview* ([5991-1910EN](#), Revision A, June 2013)
- *Manual: Agilent Metabolomics Workflow - Discovery Workflow Guide* ([5990-7067EN](#), Revision B, October 2012)
- *Manual: Agilent Metabolomics Workflow - Discovery Workflow Overview* ([5990-7069EN](#), Revision B, October 2012)
- *Manual: Agilent G3835AA MassHunter Mass Profiler Professional - Quick Start Guide* ([G3835-90009](#), Revision A, November 2012)
- *Manual: Agilent G3835AA MassHunter Mass Profiler Professional - Familiarization Guide* ([G3835-90010](#), Revision A, November 2012)
- *Manual: Agilent G3835AA MassHunter Mass Profiler Professional - Application Guide* ([G3835-90011](#), Revision A, November 2012)
- *Presentation: Advances in Instrumentation and Software for Metabolomics Research* ([Advances in Instrumentation and Software for Metabolomics.pdf](#), September 18, 2012)
- *Presentation: Two Workflows to Support Automated Class Prediction with Complex Samples* ([WP20_405_Two_Workflows_Support_Automated_Class_Prediction.pdf](#), June 25, 2012)
- *Presentation: Predictive Classification of Contaminants Encountered During the Distillation of Shochu, a Distilled Beverage Native to Japan* ([ASMS_2011_ThP_316.pdf](#), June 23, 2011)
- *Brochure: Agilent Solutions for Metabolomics* ([5990-6048EN](#), April 30, 2012)
- *Brochure: Agilent Mass Profiler Professional Software* ([5990-4164EN](#), April 27, 2012)
- *Application: Detecting Contamination in Shochu Using the Agilent GC/MSD, Mass Profiler Professional, and Sample Class Prediction Models* ([5991-0975EN](#), August 2, 2012)
- *Application: Metabolomic Profiling of Wines using LC/QTOF MS and MassHunter Data Mining and Statistical Tools* ([5990-8451EN](#), June 22, 2011)

A complete list of references may be found in "References" in the *Class Prediction with Agilent Mass Profiler Professional - Workflow Guide*.



www.agilent.com

© Agilent Technologies, Inc. 2014
Revision A, August 2014



5991-1912EN



Agilent Technologies