* [Classifying Cell Samples (SVM)](http://127.0.0.1:54857/help/topic/com.ibm.spss.modeler.tutorial/clementine/example_svm_intro.htm)

# Classifying Cell Samples (SVM)

Support Vector Machine (SVM) is a classification and regression technique that is particularly suitable for wide datasets. A wide dataset is one with a large number of predictors, such as might be encountered in the field of bioinformatics (the application of information technology to biochemical and biological data).

A medical researcher has obtained a dataset containing characteristics of a number of human cell samples extracted from patients who were believed to be at risk of developing cancer. Analysis of the original data showed that many of the characteristics differed significantly between benign and malignant samples. The researcher wants to develop an SVM model that can use the values of these cell characteristics in samples from other patients to give an early indication of whether their samples might be benign or malignant.

This example uses the stream named svm\_cancer.str, available in the Demos folder under the streams subfolder. The data file is cell\_samples.data. See the topic [Demos Folder](http://127.0.0.1:54857/help/topic/com.ibm.spss.modeler.tutorial/clementine/entities/demo_folder_intro.htm#demo_folder_intro) for more information.

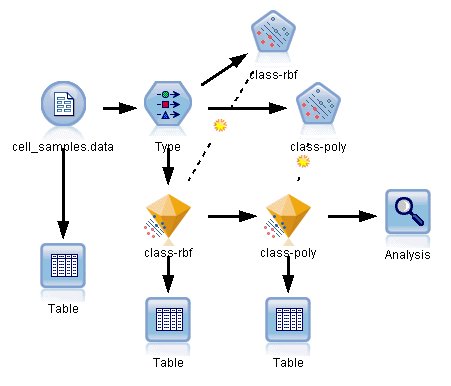
The example is based on a dataset that is publicly available from the UCI Machine Learning Repository . The dataset consists of several hundred human cell sample records, each of which contains the values of a set of cell characteristics. The fields in each record are:

| **Field name** | **Description** |
| --- | --- |
| ID | Patient identifier |
| Clump | Clump thickness |
| UnifSize | Uniformity of cell size |
| UnifShape | Uniformity of cell shape |
| MargAdh | Marginal adhesion |
| SingEpiSize | Single epithelial cell size |
| BareNuc | Bare nuclei |
| BlandChrom | Bland chromatin |
| NormNucl | Normal nucleoli |
| Mit | Mitoses |
| Class | Benign or malignant |

For the purposes of this example, we're using a dataset that has a relatively small number of predictors in each record.

**Creating the Stream**

*Figure 1. Sample stream to show SVM modeling*

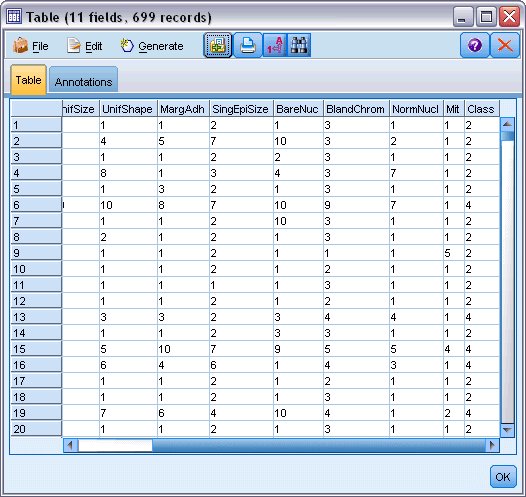


1. Create a new stream and add a Var File source node pointing to *cell\_samples.data* in the *Demos* folder of your IBM® SPSS® Modeler installation.

Let's take a look at the data in the source file.

1. Add a Table node to the stream.
2. Attach the Table node to the Var File node and run the stream.

*Figure 2. Source data for SVM*



The *ID* field contains the patient identifiers. The characteristics of the cell samples from each patient are contained in fields *Clump* to *Mit*. The values are graded from 1 to 10, with 1 being the closest to benign.

The *Class* field contains the diagnosis, as confirmed by separate medical procedures, as to whether the samples are benign (value = 2) or malignant (value = 4).

*Figure 3. Type node settings*



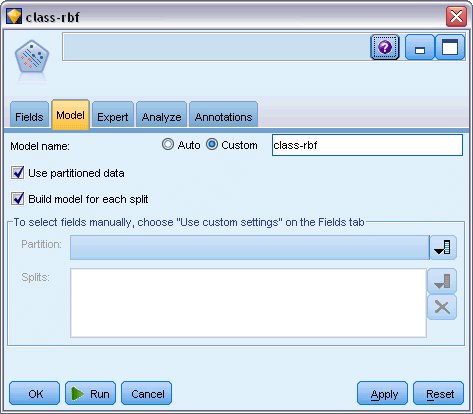
1. Add a Type node and attach it to the Var File node.
2. Open the Type node.

We want the model to predict the value of *Class* (that is, benign (=2) or malignant (=4)). As this field can have one of only two possible values, we need to change its measurement level to reflect this.

1. In the **Measurement** column for the *Class* field (the last one in the list), click the value **Continuous** and change it to **Flag**.
2. Click **Read Values**.
3. In the **Role** column, set the role for *ID* (the patient identifier) to **None**, as this will not be used either as a predictor or a target for the model.
4. Set the role for the target, *Class*, to **Target** and leave the role of all the other fields (the predictors) as **Input**.
5. Click **OK**.

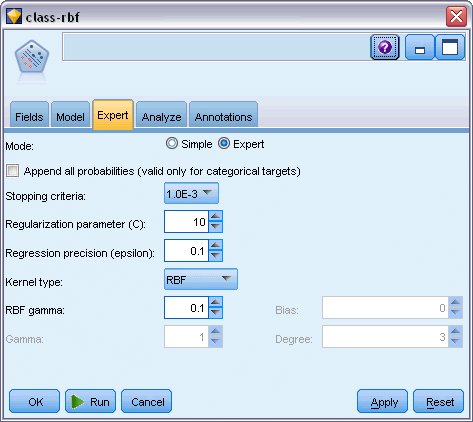
The SVM node offers a choice of kernel functions for performing its processing. As there's no easy way of knowing which function performs best with any given dataset, we'll choose different functions in turn and compare the results. Let's start with the default, RBF (Radial Basis Function).

*Figure 4. Model tab settings*



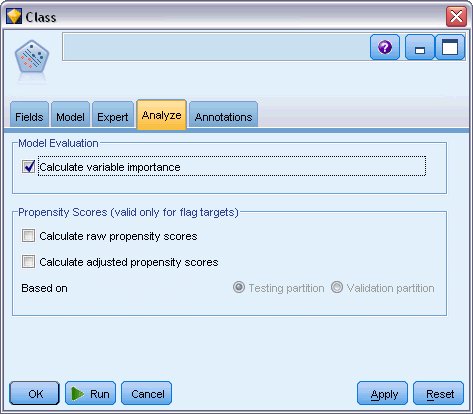
1. From the Modeling palette, attach an SVM node to the Type node.
2. Open the SVM node. On the **Model** tab, click the **Custom** option for **Model name** and type *class-rbf* in the adjacent text field.

*Figure 5. Default Expert tab settings*



1. On the **Expert** tab, set the **Mode** to **Expert** for readability but leave all the default options as they are. Note that **Kernel type** is set to **RBF** by default. All the options are greyed out in Simple mode.

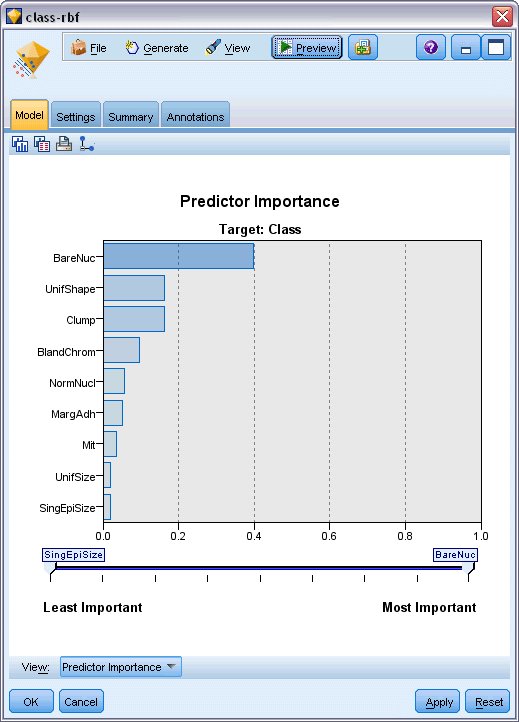
*Figure 6. Analyze tab settings*



1. On the **Analyze** tab, select the **Calculate variable importance** check box.
2. Click **Run**. The model nugget is placed in the stream, and in the Models palette at the top right of the screen.
3. Double-click the model nugget in the stream.

**Examining the Data**

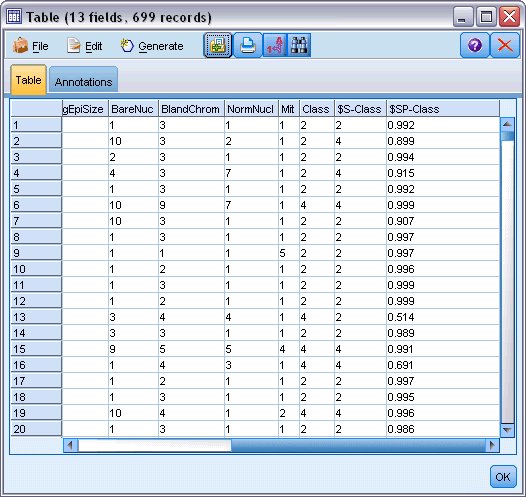
*Figure 1. Predictor Importance graph*



On the Model tab, the Predictor Importance graph shows the relative effect of the various fields on the prediction. This shows us that *BareNuc* has easily the greatest effect, while *UnifShape* and *Clump* are also quite significant.

1. Click **OK**.
2. Attach a Table node to the *class-rbf* model nugget.
3. Open the Table node and click **Run**.

*Figure 2. Fields added for prediction and confidence value*



1. The model has created two extra fields. Scroll the table output to the right to see them:

| **New field name** | **Description** |
| --- | --- |
| *$S-Class* | Value for *Class* predicted by the model. |
| *$SP-Class* | Propensity score for this prediction (the likelihood of this prediction being true, a value from 0.0 to 1.0). |

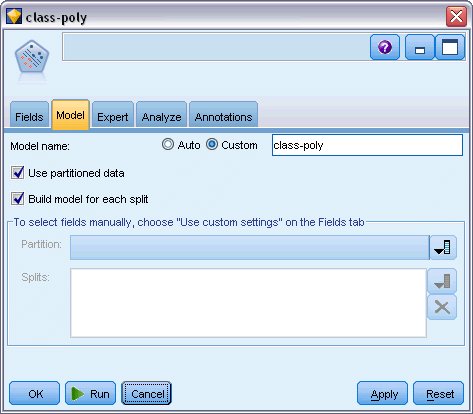
Just by looking at the table, we can see that the propensity scores (in the *$SP-Class* column) for most of the records are reasonably high.

However, there are some significant exceptions; for example, the record for patient 1041801 at line 13, where the value of 0.514 is unacceptably low. Also, comparing *Class* with *$S-Class*, it's clear that this model has made a number of incorrect predictions, even where the propensity score was relatively high (for example, lines 2 and 4).

Let's see if we can do better by choosing a different function type.

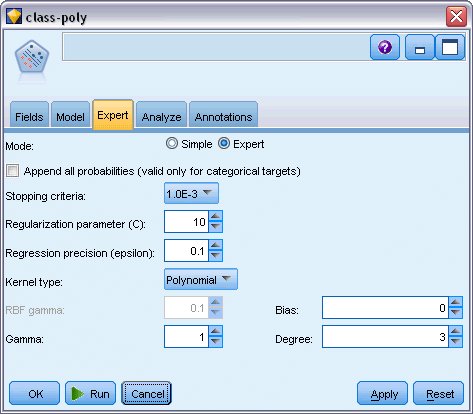
**Trying a Different Function**

*Figure 1. Setting a new name for the model*



1. Close the Table output window.
2. Attach a second SVM modeling node to the Type node.
3. Open the new SVM node.
4. On the **Model** tab, choose Custom and type *class-poly* as the model name.

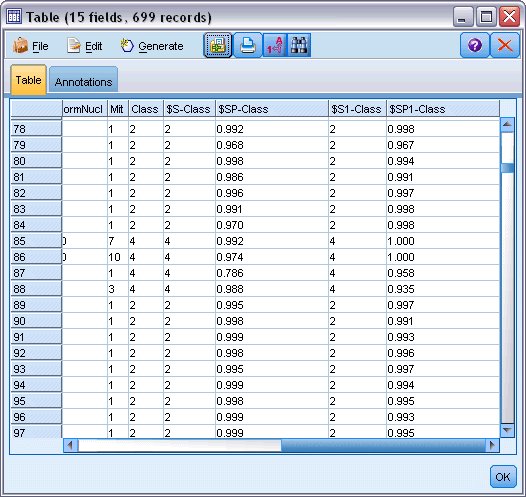
*Figure 2. Expert tab settings for Polynomial*



1. On the **Expert** tab, set **Mode** to **Expert**.
2. Set **Kernel type** to **Polynomial** and click **Run**. The *class-poly* model nugget is added to the stream, and also to the Models palette at the top right of the screen.
3. Connect the *class-rbf* model nugget to the *class-poly* model nugget (choose **Replace** at the warning dialog).
4. Attach a Table node to the *class-poly* nugget.
5. Open the Table node and click **Run**.

**Comparing the Results**

*Figure 1. Fields added for Polynomial function*



1. Scroll the table output to the right to see the newly added fields.

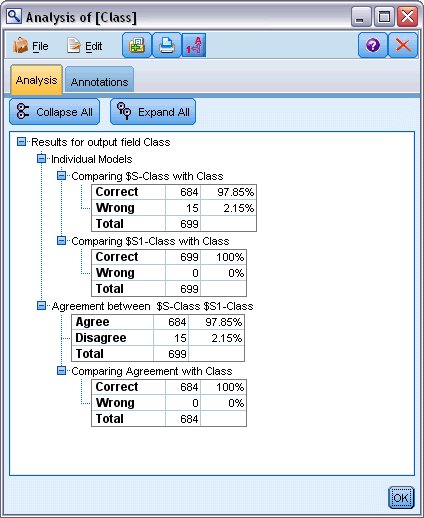
The generated fields for the Polynomial function type are named *$S1-Class* and *$SP1-Class*.

The results for Polynomial look much better. Many of the propensity scores are 0.995 or better, which is very encouraging.

1. To confirm the improvement in the model, attach an Analysis node to the *class-poly* model nugget.

Open the Analysis node and click **Run**.

*Figure 2. Analysis node*



This technique with the Analysis node enables you to compare two or more model nuggets of the same type. The output from the Analysis node shows that the RBF function correctly predicts 97.85% of the cases, which is still quite good. However, the output shows that the Polynomial function has correctly predicted the diagnosis in every single case. In practice you are unlikely to see 100% accuracy, but you can use the Analysis node to help determine whether the model is acceptably accurate for your particular application.

In fact, neither of the other function types (Sigmoid and Linear) performs as well as Polynomial on this particular dataset. However, with a different dataset, the results could easily be different, so it's always worth trying the full range of options.

# Summary

You have used different types of SVM kernel functions to predict a classification from a number of attributes. You have seen how different kernels give different results for the same dataset and how you can measure the improvement of one model over another.