GeneXproTools Logistic Regression Help

GeneXproTools 4.0

Modeling made easy

Logistic Regression Analytics Platform

GeneXproTools Editions | Learning Resources | Support | Contact Us

◁ | ▶

Logistic Regression

Introduction

The goal in **Logistic Regression** is to assign probabilities to model scores, creating a reliable **ranking system** that can be used straightaway to evaluate the risk involved in financial and insurance applications, to rank potential respondents in a marketing campaign, or to evaluate the risk of contracting a disease.

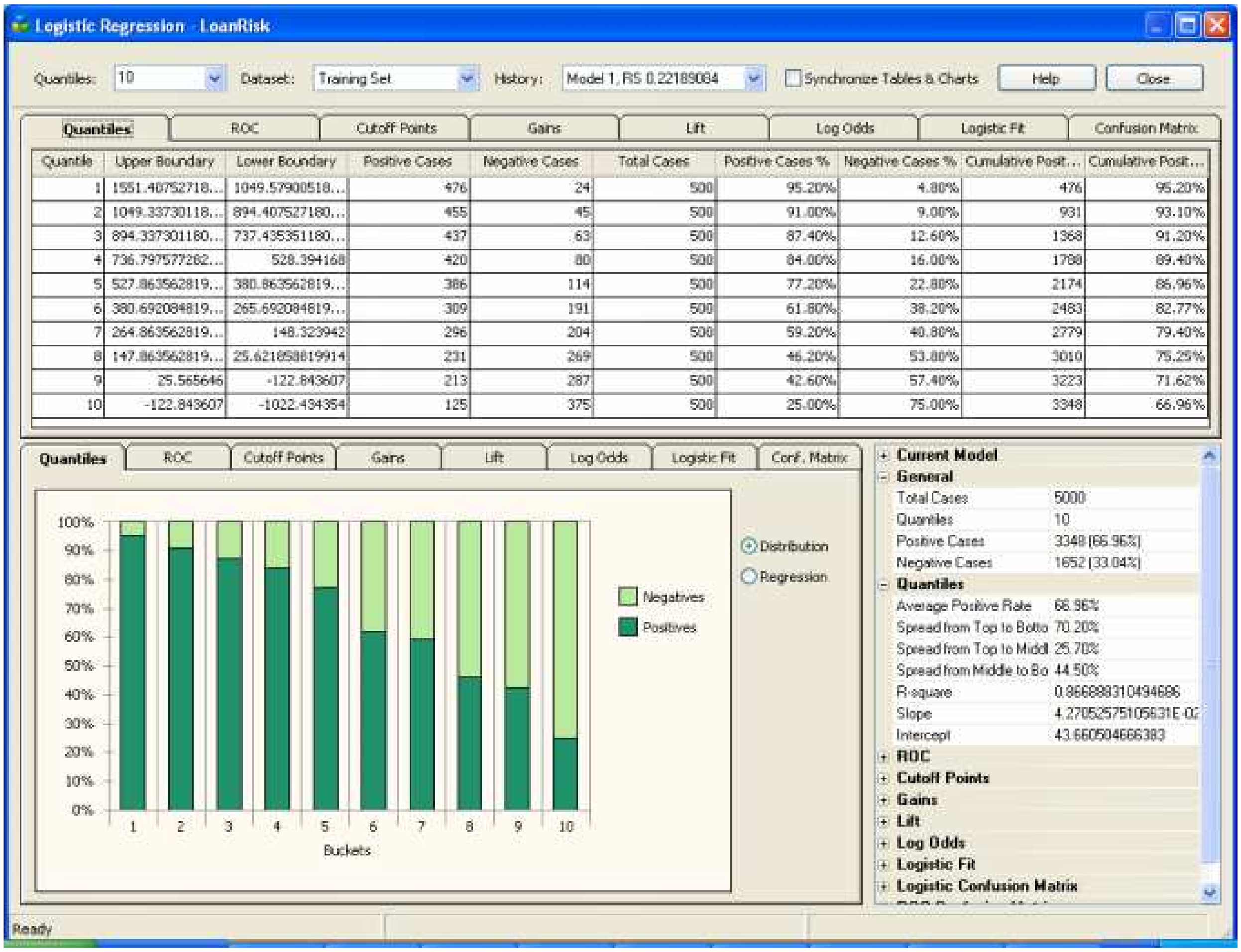
The **Logistic Regression Framework** of GeneXproTools builds on the models it generates with its evolutionary algorithms and then applies the canonical logistic regression technique to estimate probabilities for each model score. And once you know the probability of an event, you can also make categorical predictions and consequently infer easily crisp confusion matrixes.

Thus, with its new Logistic Regression Framework, GeneXproTools offers a highly **robust hybrid system** in which sophisticated multivariate nonlinear models are easily created by evolution and then empowered by traditional statistical modeling techniques.

With the Logistic Regression Framework of GeneXproTools you can:

- Perform Quantile Analysis and Regression of your model scores
- Build and analyze ROC Curves
- Determine Optimal Cutoff Points for your model scores
- Perform detailed analyses of Gains and Lift Charts
- Create powerful Logistic Regression Models

- Assign probabilities to your model scores
- Build and analyze Logistic Fit Charts
- Infer and analyze Logistic and ROC Confusion Matrixes
- Test the predictive accuracy of your models on unseen data
- Rank new cases easily using the Logistic Regression Scoring Engine
- Explore vast solution spaces with a wide set of fitness functions
- Evolve models using different class encodings
- Convert Classification runs to the Logistic Regression Platform
- Convert Logistic Regression runs to the Classification Platform



Logistic Regression - LoanRisk

Quantiles: 10 Dataset: Training Set History: Model 1, R5 0.22189084 ☐ Synchronize Tables & Charts

Quantile	Upper Boundary	Lower Boundary	Positive Cases	Negative Cases	Total Cases	Positive Cases %	Negative Cases %	Cumulative Posit...	Cumulative Posit...
1	1551.40752718...	1049.57900518...	476	24	500	95.20%	4.80%	476	95.20%
2	1049.33730118...	894.407527180...	455	45	500	91.00%	9.00%	931	93.10%
3	894.337301180...	737.435351180...	437	63	500	87.40%	12.60%	1368	91.20%
4	736.797577282...	528.394168	420	80	500	84.00%	16.00%	1788	89.40%
5	527.863562819...	380.863562819...	386	114	500	77.20%	22.80%	2174	86.96%
6	380.692084819...	265.692084819...	309	191	500	61.80%	38.20%	2483	82.77%
7	264.863562819...	148.323942	296	204	500	59.20%	40.80%	2779	79.40%
8	147.863562819...	25.621858819914	231	269	500	46.20%	53.80%	3010	75.25%
9	25.565646	-122.843607	213	287	500	42.60%	57.40%	3223	71.62%
10	-122.843607	-1022.434354	125	375	500	25.00%	75.00%	3348	66.96%

● Distribution
○ Regression

General

Total Cases	5000
Quantiles	10
Positive Cases	3348 (66.96%)
Negative Cases	1652 (33.04%)

Quantiles

Average Positive Rate	66.96%
Spread from Top to Botto	70.20%
Spread from Top to Middl	25.70%
Spread from Middle to Bo	44.50%
R-square	0.866888310494686
Slope	4.27052575105631E-02
Intercept	43.660504666383

Ready

Logistic Regression

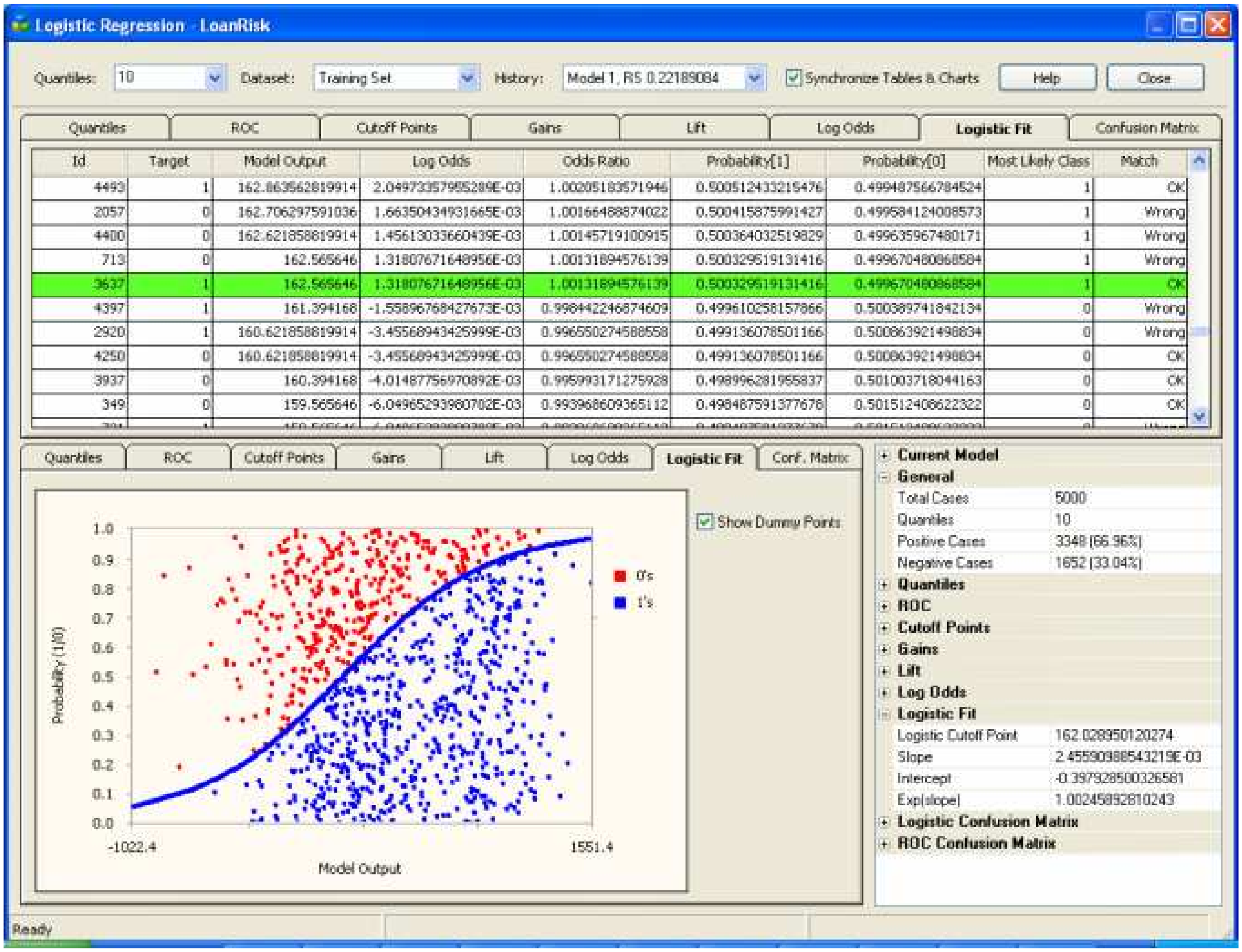
Getting Started

In order to access the Logistic Regression Framework of GeneXproTools 4.0 you need to:

1. Create a statistical model that explains a binary dependent variable, using either the Function Finding / Logistic Regression Framework or the Classification Framework of GeneXproTools.

In the **Function Finding / Logistic Regression Framework** all datasets with binary dependent variables use by default the Correlation Coefficient Fitness Function as this kind of function gives the best results with the standard 0/1 class encoding.
In the **Classification Framework** you have access to a wide variety of fitness functions that despite using pre-established rounding thresholds, offer interesting alternatives for exploring the solution space.

2. In the Function Finding / Logistic Regression Framework, click the **Logistic Regression menu** and then choose one of the available analytics tools: Quantile Analysis, ROC Curve, Cutoff Points, Gains Chart, Lift Chart, Log Odds, Logistic Fit, or Confusion Matrix.
   This activates the number-crunching process of the **Logistic Regression Analytics Platform** that starts with the construction of the Quantile Table and finishes with the creation of the Logistic Regression Model and the inference of very clear Confusion Matrixes. After the processing has stopped, you just have to navigate from tab to tab to evaluate the quality and performance of your modeling system.



Quantiles: 10 Dataset: Training Set History: Model 1, RS 0.22189084 ☑ Synchronize Tables & Charts

Id	Target	Model Output	Log Odds	Odds Ratio	Probability[1]	Probability[0]	Most Likely Class	Match
4493	1	162.863562819914	2.04973357955289E-03	1.00205183571946	0.500512433215476	0.499487566784524	1	OK
2057	0	162.706297591036	1.66350434931665E-03	1.00166488874022	0.500415875991427	0.499584124008573	1	Wrong
4400	0	162.621858819914	1.45613033660439E-03	1.00145719100915	0.500364032519829	0.499635967480171	1	Wrong
713	0	162.565646	1.31807671648956E-03	1.00131894576139	0.500329519131416	0.499670480868584	1	Wrong
3637	1	162.565646	1.31807671648956E-03	1.00131894576139	0.500329519131416	0.499670480868584	1	OK
4397	1	161.394168	-1.55896768427673E-03	0.998442246874609	0.499610258157866	0.500389741842134	0	Wrong
2920	1	160.621858819914	-3.45568943425999E-03	0.996550274588558	0.499136078501166	0.500863921498834	0	Wrong
4250	0	160.621858819914	-3.45568943425999E-03	0.996550274588558	0.499136078501166	0.500863921498834	0	OK
3937	0	160.394168	-4.01487756970892E-03	0.995993171275928	0.498996281955837	0.501003718044163	0	OK
349	0	159.565646	-6.04965293980702E-03	0.993968609365112	0.498487591377678	0.501512408622322	0	OK

☑ Show Dummy Points

- ⊞ **Current Model**
- ⊟ **General**

Total Cases	5000
Quantiles	10
Positive Cases	3348 (66.96%)
Negative Cases	1652 (33.04%)

- ⊞ **Quantiles**
- ⊞ **ROC**
- ⊞ **Cutoff Points**
- ⊞ **Gains**
- ⊞ **Lift**
- ⊞ **Log Odds**
- ⊟ **Logistic Fit**

Logistic Cutoff Point	162.028950120274
Slope	2.45590988543219E-03
Intercept	-0.397928500326581
Exp(slope)	1.00245892810243

- ⊞ **Logistic Confusion Matrix**
- ⊞ **ROC Confusion Matrix**

In the Logistic Regression Window of GeneXproTools you can:

1. Analyze and create Quantile Tables and Charts; perform Quantile Regression; analyze the ROC Curve of your models; visualize the Optimal Cutoff Point for your test scores; study the Gains and Lift Charts of your models; access the Log Odds Chart used to evaluate the slope and intercept of the Logistic Regression Model; visualize how well your logistic model fits the data in the Logistic Fit Chart; and compare and analyze Logistic and ROC Confusion Matrixes using both 2 x 2 Contingency Tables and quantile-based Distribution Charts.
2. Copy all the Tables and Charts to the clipboard.
   All the Tables and Charts generated within the Logistic Regression Window can be copied to the

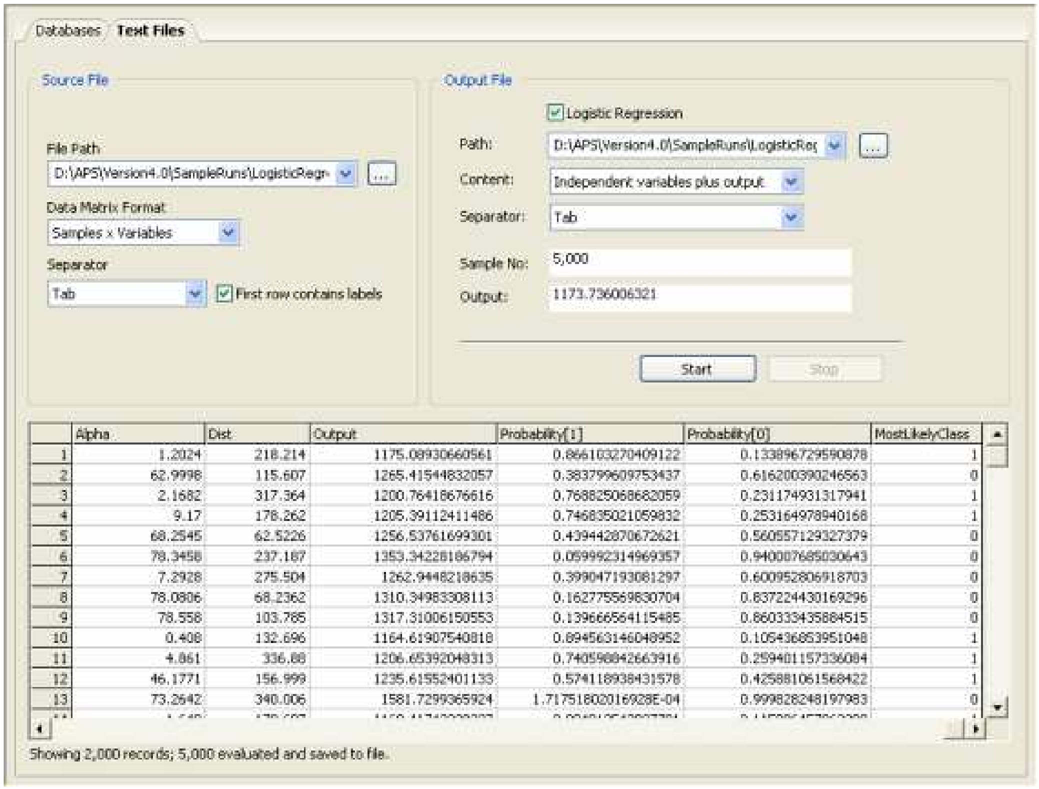
clipboard through the mouse right-click menu. Tables can be copied in their entirety or you can copy just selected rows or individual columns.

3. Print all the Charts.
   All the Charts in the Logistic Regression Window (Quantile Charts, ROC Curve, Cutoff Points, Gains Chart, Lift Chart, Log Odds, Logistic Fit, and Confusion Matrixes) can be printed through the mouse right-click menu.
4. Copy and print the Statistics Report.
   The Stats Report summarizes all the relevant parameters and statistics derived from all the analyses (Quantile Regression, ROC Curve, Cutoff Points, Gains Chart, Lift Chart, Log Odds, Logistic Fit, and Confusion Matrixes) performed for the active model and active dataset. It also contains relevant information about the training and testing data, such as class distribution and number of records. And finally, the Stats Report also summarizes some basic information about the model, such as its fitness and R-square and if any calculation errors occurred during the computation of the model scores. Within the Logistic Regression Window, all such calculation errors (they can happen when processing the unseen data of the testing set and also of the "training dataset" if it was replaced by another one or if the model itself was modified by the user) return zero so that the calculations can resume. Note, however, that GeneXproTools flags these errors clearly, highlighting them in light red in all the tables where the model outputs are shown (ROC Table, Cutoff Points Table, Logistic Fit Table, and Confusion Matrix Table).
5. Choose a different number of buckets for your Quantile Table and then see immediately how it affects the Logistic Regression Model through the Logistic Fit Chart.
   The number of buckets is an essential parameter for most of the analyses performed in the Logistic Regression Window (Quantile Regression, Gains Chart, Lift Chart, Log Odds and Logistic Regression, Logistic Fit, and Logistic Confusion Matrix) and therefore it is saved for each model and for each dataset.
   By using the ROC-derived accuracy as your golden standard (it is quantile-independent and remains unchanged for a particular model), you can fine-tune the number of buckets to get the most of your models. Note, however, that it is not uncommon to get better performance on the Logistic Confusion Matrix, which of course is indicative of a very good Logistic Fit.
6. Access the **testing dataset** so that you can not only test further the **predictive accuracy** of your model but also build logistic regression models with it.
   The testing dataset was never brought into contact with the model during the training process and therefore constitutes an excellent blind test for checking the predictive accuracy of your model on unseen data. You access the testing dataset by choosing Testing Set in the Dataset combo box. GeneXproTools then creates a specific Quantile Table for the testing dataset and also performs the complete logistic regression analysis for this dataset. Note, however, that if you want to use this logistic regression model (that is, the slope and intercept evaluated for the testing set) for scoring new cases using the Scoring Engine of GeneXproTools, you'll have to replace the original training dataset with this one and then recalculate the logistic parameters (the slope and intercept of the Log Odds Chart) with this new operational dataset.
7. Analyze all the **intermediate models** created in a run by selecting any model in the History combo box.
   Each model in the Run History is identified by its ID and training R-square for easy access in the History combo box (although controversial, the small R-square values typical of risk assessment and response models are useful indicators of the quality of a model and are in fact widely used by real-world modelers; for instance, R-square values around 0.23 are considered excellent for typical risk assessment models and indicative of a good fit). Note that when you close the Logistic Regression Window, the last observed model will remain your active model.
   Although modelers are understandably interested in the best-of-run model, it's great fun to get a glimpse of how evolution works by being able to see how intermediate models behave and how their performance becomes better and better with time. But this process is also important to develop a good intuition and learn some tips that might prove useful to better guide the evolutionary process.
8. Choose to browse all the available tables and charts in synchrony or asynchronously by ticking the Synchronize Tables & Charts check-box.
   By default, the Tables & Charts of the Logistic Regression Framework of GeneXproTools move in synchrony. But you can have them move independently so that you can look at any of the tables while analyzing a certain chart and vice versa. Another advantage of having Tables & Charts move independently is that it's much quicker to move from chart to chart when using very large datasets.
9. Access the **Logistic Regression Help File**.

By clicking the Help button a new window with the Logistic Regression Help File pops up so that you can access any section in the Logistic Regression Documentation to answer any question you might have regarding any of the analyses of the Logistic Regression Platform. For help on general evolutionary model design, you'll have to consult the main GeneXproTools Help File.

In order to make predictions or rank new cases within GeneXproTools, you need to:

1. Choose and evaluate all the **key parameters** (number of buckets plus the slope and intercept for the logistic regression model) in the Logistic Regression Window.
   You start by selecting the model you are interested in (usually the best-of-run model) and then enter the Logistic Regression Window by choosing one of the entry points listed under the Logistic Regression menu (Quantile Analysis, ROC Curve, Cutoff Points, Gains Chart, Lift Chart, Log Odds, Logistic Fit, and Confusion Matrix). GeneXproTools executes all the required calculations automatically, although you may want to fine-tune the number of quantiles to achieve the best possible performance with the Logistic Regression Model.
2. Close the Logistic Regression Window and then go to the **Scoring Panel**.
   To score a database or Excel file, on the Scoring menu select Databases or go to the Scoring Panel and select the Databases Tab. For scoring data kept in text files, on the Scoring menu select Text Files or go to the Scoring Panel and select the Text Files Tab.
3. In the Scoring Panel tick the Logistic Regression check-box and then enter the path for both the source data and output file.
   The Scoring Engine of GeneXproTools uses the **Javascript code** of your model to perform the computations as it already contains the code for the UDFs and DDFs.
4. Then press the Start button to begin the scoring process.
   GeneXproTools saves the **scoring results** to a file which contains the predictions of your model for all the new cases in the source file. For small datasets (up to 16 variables and 2000 cases) GeneXproTools also shows the scoring results in the table of the Scoring Panel. Besides the model output, GeneXproTools also computes the probabilities of being either a “1” or “0” and also infers the most likely class for all the cases.



File Path: D:\APS\Version4.0\SampleRuns\LogisticRegr

Data Matrix Format: Samples x Variables

Separator: Tab

☑ First row contains labels

☑ Logistic Regression

Path: D:\APS\Version4.0\SampleRuns\LogisticReg

Content: Independent variables plus output

Separator: Tab

Sample No: 5,000

Output: 1173.736006321

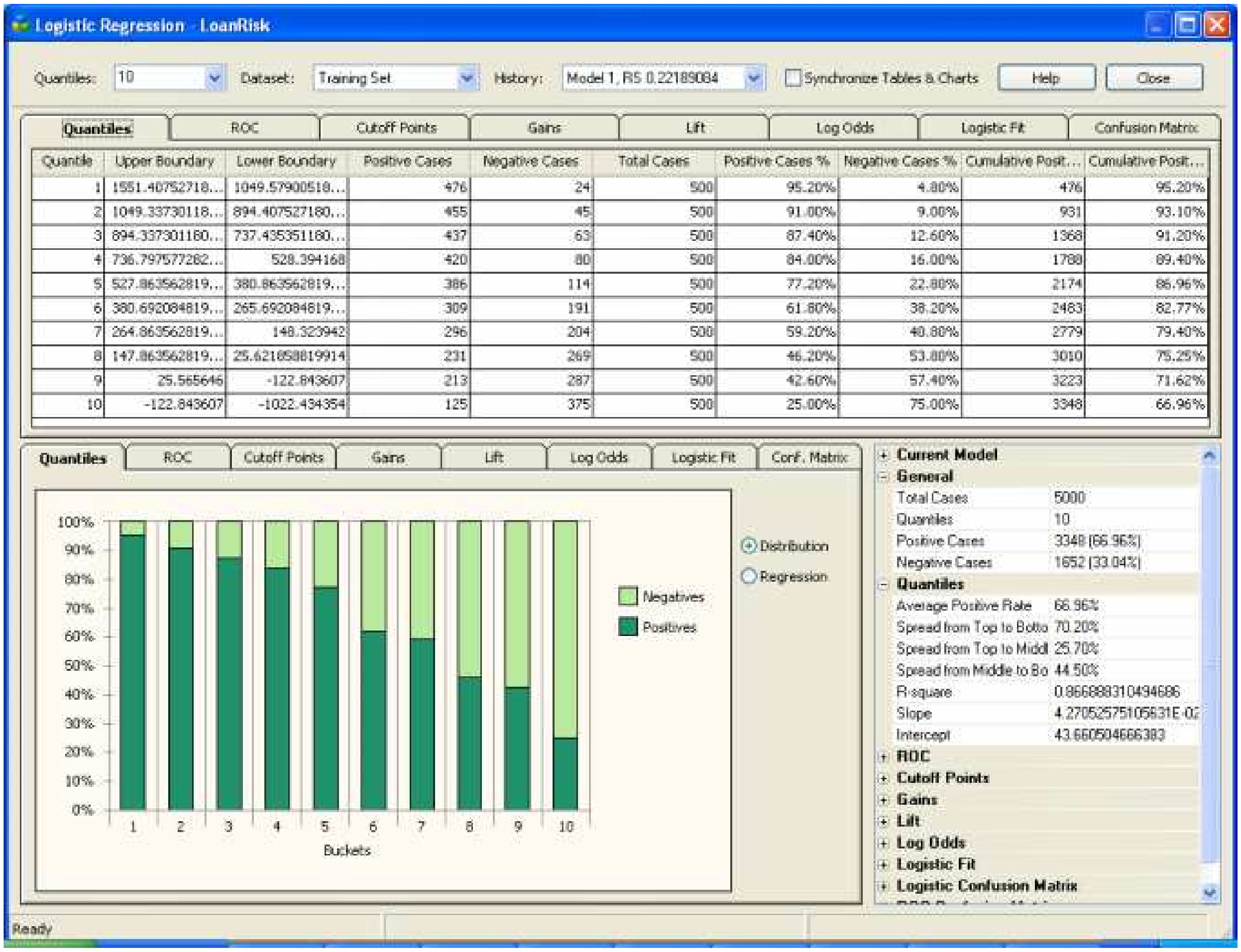
	Alpha	Dist	Output	Probability[1]	Probability[0]	MostLikelyClass
1	1.2024	218.214	1175.08930660561	0.866103270409122	0.133896729590878	1
2	62.9998	115.607	1265.41544832057	0.383799609753437	0.616200390246563	0
3	2.1682	317.364	1200.76418676616	0.768825068682059	0.231174931317941	1
4	9.17	178.262	1205.39112411486	0.746835021059832	0.253164978940168	1
5	68.2545	62.5226	1256.53761699301	0.439442870672621	0.560557129327379	0
6	78.3458	237.187	1353.34228186794	0.059992314969357	0.940007685030643	0
7	7.2928	275.504	1262.9448218635	0.399047193081297	0.600952806918703	0
8	78.0806	68.2362	1310.34983308113	0.162775569830704	0.837224430169296	0
9	78.558	103.785	1317.31006150553	0.139666564115485	0.860333435884515	0
10	0.408	132.696	1164.61907540818	0.894563146048952	0.105436853951048	1
11	4.861	336.88	1206.65392048313	0.740598842663916	0.259401157336084	1
12	46.1771	156.999	1235.61552401133	0.574118938431578	0.425881061568422	1
13	73.2642	340.006	1581.7299365924	1.71751802016928E-04	0.999828248197983	0

Showing 2,000 records; 5,000 evaluated and saved to file.

Logistic Regression

Quantile Analysis and Regression

Quantile Tables are by themselves powerful analytics tools, but they are also at the heart of the Logistic Regression Model and Logistic Fit. In addition, they are also the basis of popular analysis tools such as Gains and Lift Charts, which are essential for making good decisions about the quality of a model and to estimate the benefits of using a model.



Quantiles: 10 Dataset: Training Set History: Model 1, R5 0.22189084 □ Synchronize Tables & Charts

Quantile	Upper Boundary	Lower Boundary	Positive Cases	Negative Cases	Total Cases	Positive Cases %	Negative Cases %	Cumulative Posit...	Cumulative Posit...
1	1551.40752718...	1049.57900518...	476	24	500	95.20%	4.80%	476	95.20%
2	1049.33730118...	894.407527180...	455	45	500	91.00%	9.00%	931	93.10%
3	894.337301180...	737.435351180...	437	63	500	87.40%	12.60%	1368	91.20%
4	736.797577282...	528.394168	420	80	500	84.00%	16.00%	1788	89.40%
5	527.863562819...	380.863562819...	386	114	500	77.20%	22.80%	2174	86.96%
6	380.692084819...	265.692084819...	309	191	500	61.80%	38.20%	2483	82.77%
7	264.863562819...	148.323942	296	204	500	59.20%	40.80%	2779	79.40%
8	147.863562819...	25.621858819914	231	269	500	46.20%	53.80%	3010	75.25%
9	25.565646	-122.843607	213	287	500	42.60%	57.40%	3223	71.62%
10	-122.843607	-1022.434354	125	375	500	25.00%	75.00%	3348	66.96%

- Current Model
- General

Total Cases	5000
Quantiles	10
Positive Cases	3348 (66.96%)
Negative Cases	1652 (33.04%)

- Quantiles

Average Positive Rate	66.96%
Spread from Top to Botto	70.20%
Spread from Top to Middl	25.70%
Spread from Middle to Bo	44.50%
R-square	0.866888310494686
Slope	4.27052575105631E-02
Intercept	43.660504666383

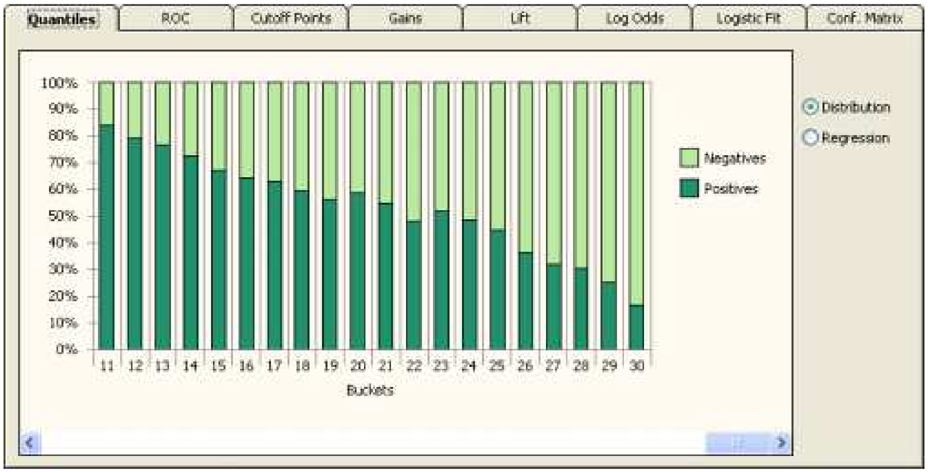
- ROC
- Cutoff Points
- Gains
- Lift
- Log Odds
- Logistic Fit
- Logistic Confusion Matrix

The number of **buckets** or **quantiles** is entered in the Quantiles combo box at the top of the Logistic Regression Window. The most commonly used Quantile Tables such as Quartiles, Quintiles, Deciles, Vingtiles, Percentiles, and 1000-tiles are listed by default, but you can type any valid quantile number in the box to build the most appropriate quantile table for your data.

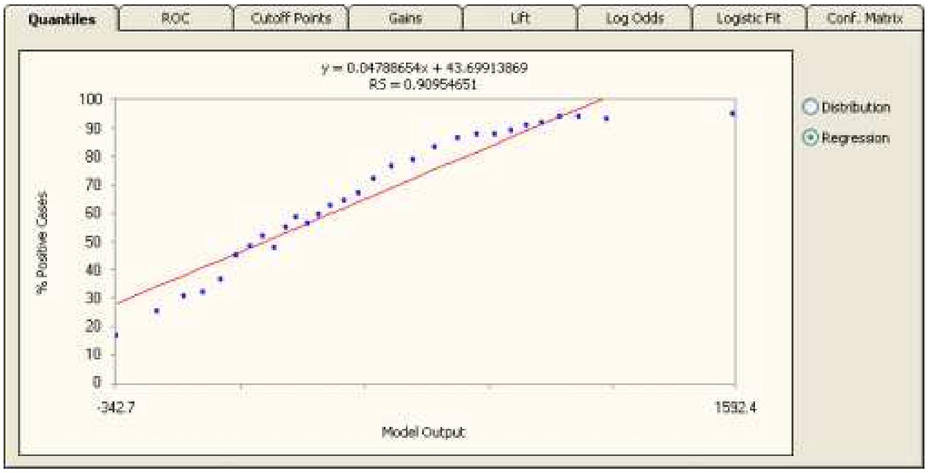
The **number of quantiles** is an essential parameter for most of the analyses performed in the Logistic Regression Window (obviously Quantile Regression and Analysis, but also Gains Chart, Lift Chart, Log Odds and Logistic Regression, Logistic Fit, and Logistic Confusion Matrix) and therefore it is saved for each model and for each dataset.

On their own, Quantile Tables are widely used in risk assessment applications and in a variety of response models to create **rankings** or **scores**. Percentiles, for instance, are very popular and often used for that purpose alone. But in GeneXproTools, Quantile Tables are explored to the fullest and are therefore also used to create a more sophisticated ranking system: the **probabilistic ranking system** of the Logistic Regression Model. This model estimates unique probabilities for each and every new case, forming a very elegant ranking system, perfectly bounded between 0 and 1.

GeneXproTools shows its Quantile Tables in **100% stacked column charts**, where the distribution of both Positive and Negative categories is shown for all the buckets. By moving the cursor over each column, GeneXproTools shows both the percentage and absolute values for each class. For more than 20 buckets, a scroll bar appears at the bottom of the **Quantile Chart** and by moving it you can see the distribution over all the range of model outputs.



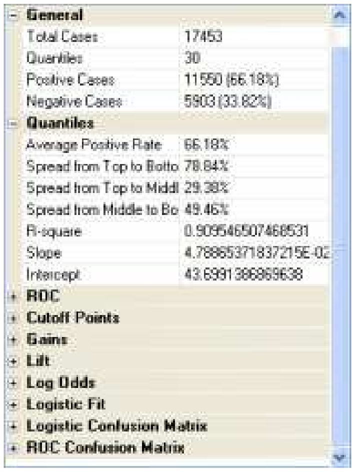
Besides allowing the visualization of Quantile Tables, GeneXproTools also shows and performs a weighted **Quantile Regression**. Both the slope and intercept of the regression line, as well as the R-square, are computed and shown in the **Quantile Regression Chart**.



These parameters form the core of the **Quantile Regression Model** and can be used both to evaluate **rankings** and to make **discrete classifications** in a fashion similar to what is done with the Logistic Regression Model. Within the Logistic Regression Framework of GeneXproTools, however, only the Logistic Regression Model is used to evaluate rankings (probabilities, in this case) and to estimate the most likely class. Furthermore, the Scoring Engine of GeneXproTools also uses the Logistic Regression Model to make predictions, not the Quantile Regression Model.

Note also that in the X-axis of the **Quantile Regression Chart**, GeneXproTools plots model outputs and therefore you can see clearly how spread out model scores are. Note also that, in the Quantile Regression Chart, upper boundaries are used if the predominant class is "1" and the model is normal, or the predominant class is "0" and the model is inverted; and lower boundaries are used if the predominant class is "1" and the model is inverted, or the predominant class is "0" and the model is normal.

On the companion **Statistics Report** on the right (the Quantiles section opens up every time the Quantiles Chart Tab is selected), GeneXproTools also shows the Spread from Top to Bottom, Spread from Top to Middle, and Spread from Middle to Bottom (when the number of buckets is even, the middle value is the average of the two middle buckets). Note that negative values for the spreads, especially the Spread from Top to Bottom, are usually indicative of an inverted model. In absolute terms, however, the wider the spread the better the model.



− **General**	
Total Cases	17453
Quantiles	30
Positive Cases	11550 (66.18%)
Negative Cases	5903 (33.82%)
− **Quantiles**	
Average Positive Rate	66.18%
Spread from Top to Botto	78.84%
Spread from Top to Middl	29.38%
Spread from Middle to Bo	49.46%
R-square	0.909546507468531
Slope	4.78865371837215E-02
Intercept	43.6991386869638
+ **ROC**	
+ **Cutoff Points**	
+ **Gains**	
+ **Lift**	
+ **Log Odds**	
+ **Logistic Fit**	
+ **Logistic Confusion Matrix**	
+ **ROC Confusion Matrix**	

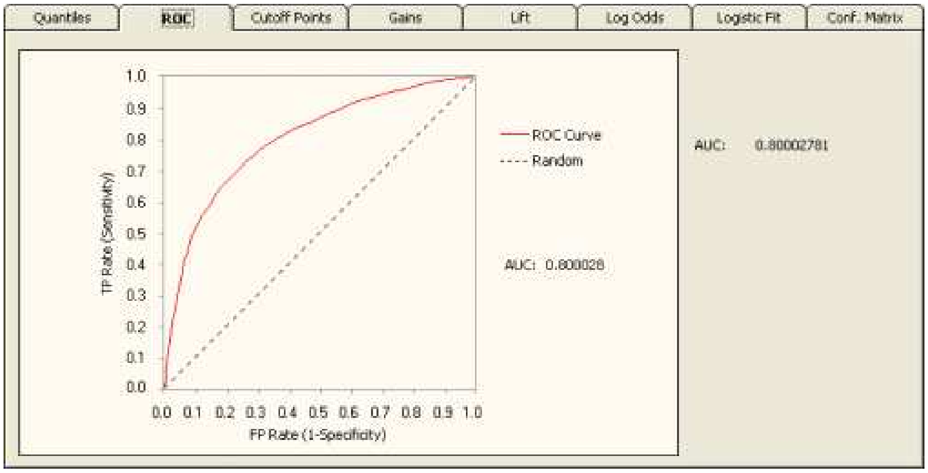
◀ | ▶

Logistic Regression

ROC Analysis

Receiver Operating Characteristic or **ROC Curves** are useful visualization tools that allow a quick assessment of the quality of a model. They are usually plotted in reference to a Baseline or Random Model, with the **Area Under the ROC Curve** (or **AUC** for short) as a popular indicator of the quality of a model.

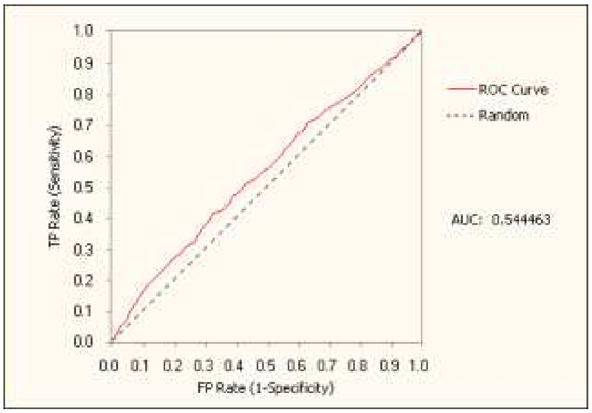
So, for the **Random Model**, the area under the ROC curve is equal to 0.5, which means that the further up (or down, for inverted models) a model is from 0.5 the better it is. Indeed, for perfect models on both sides of the random line, what is called **ROC heaven** takes place when AUC = 1 (for normal models) or AUC = 0 (for inverted models). Below is shown a typical ROC curve obtained for a risk assessment model using a training dataset with 18,253 cases. This model has an R-square of 0.245889 (R-square values might seem unusually low, but in risk assessment applications R-square values around 0.23 are considered excellent and indicative of a good model) and an AUC of 0.800028.

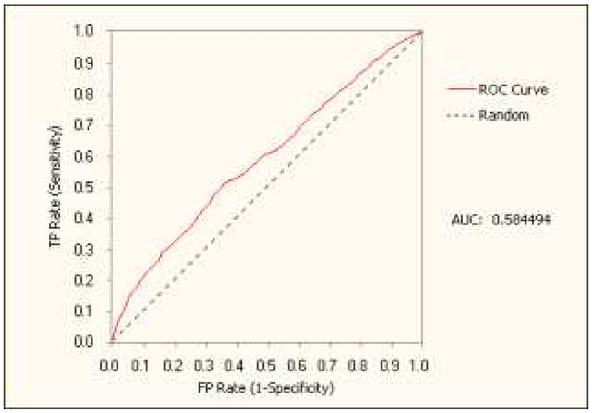


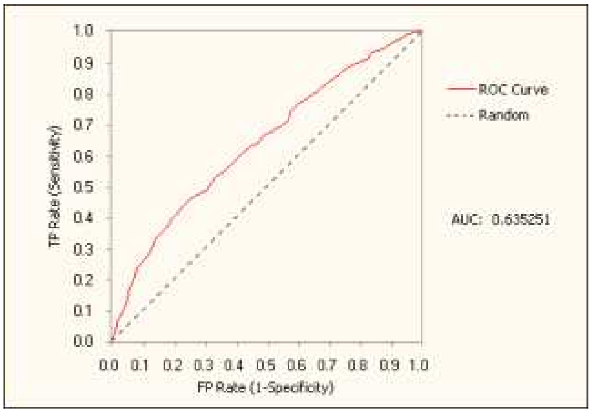
AUC: 0.80002781

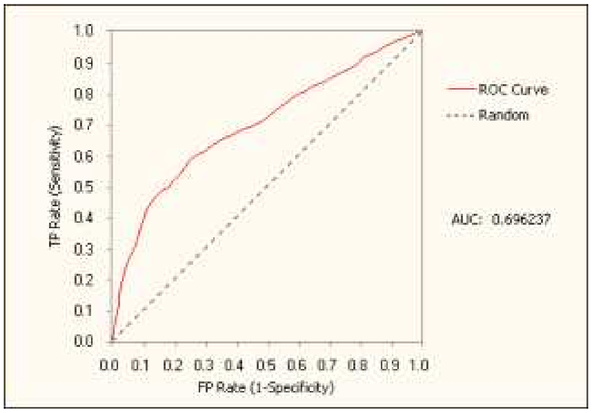
And below is shown a **Gallery of ROC Curves** typical of intermediate models generated during a GeneXproTools run. These specifically were created for a risk assessment problem with a training dataset with 18,253 cases and using a small population of just 30 programs. The R-square of each model, as well as the generation at which they were discovered, are also shown as illustration. From left to right and top to bottom, they are as follow (see also the twin Gallery of Logistic Fit Charts in the Logistic Fit section):

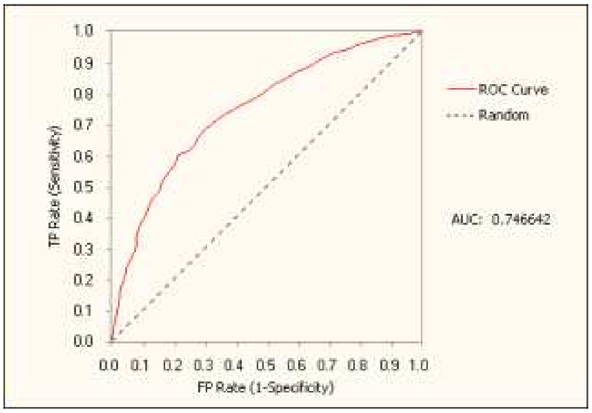
- Generation 0, R-square = 0.002221, AUC = 0.544463
- Generation 3, R-square = 0.022389, AUC = 0.584494
- Generation 8, R-square = 0.050686, AUC = 0.635251
- Generation 12, R-square = 0.064736, AUC = 0.696237
- Generation 14, R-square = 0.163695, AUC = 0.746642
- Generation 344, R-square = 0.219212, AUC = 0.782164

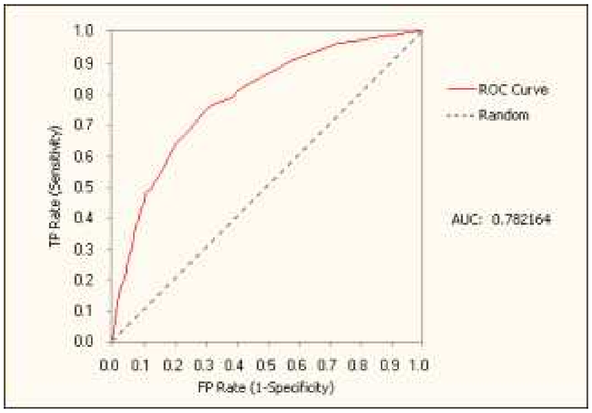










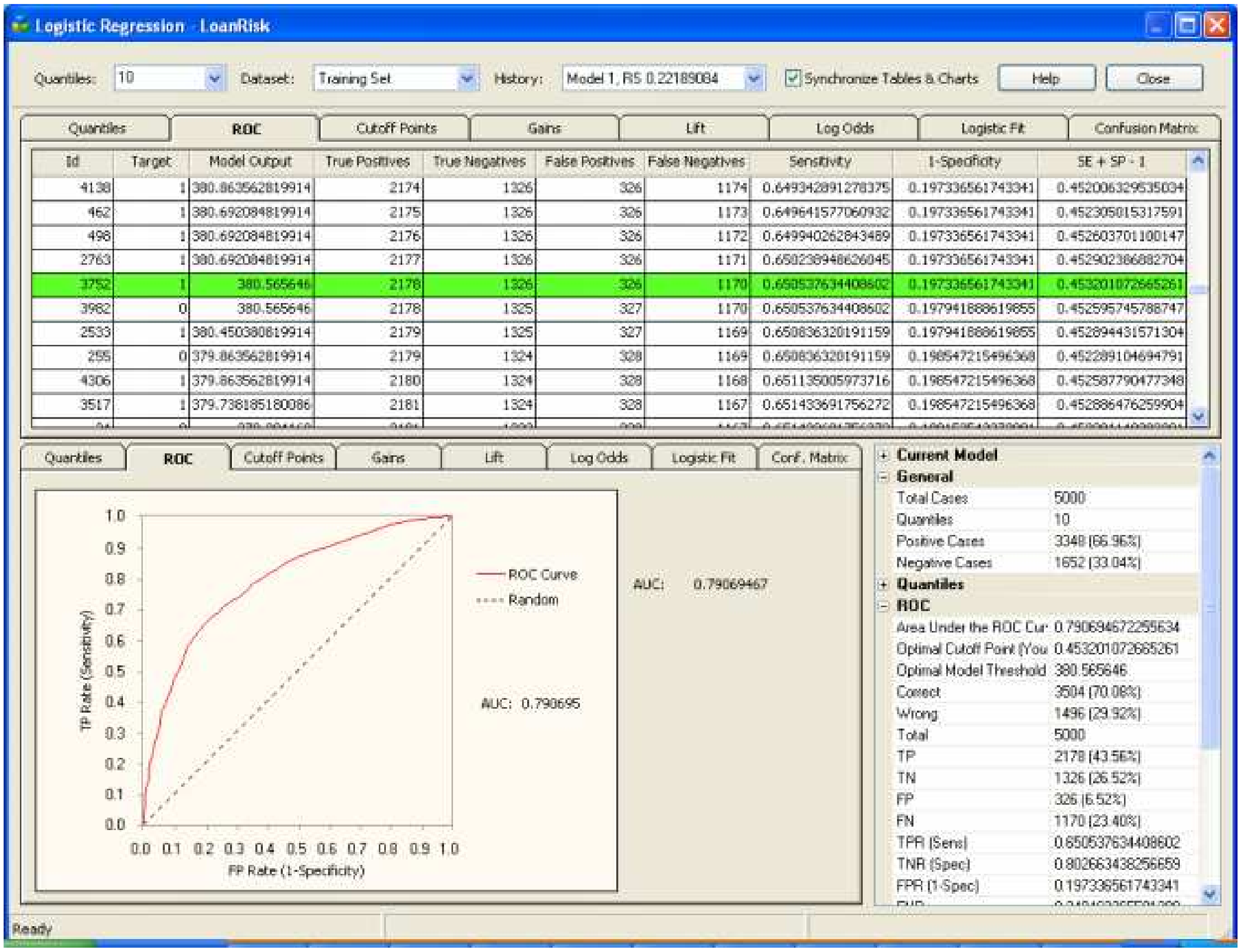


ROC Curves and Tables are also useful to evaluate what is called the **Optimal Cutoff Point**, which is given by the maximum of the **Youden index**. The Youden index *J* returns the maximum value of the expression (for inverted models, it returns the minimum):

$$J = \max[SE_{(t)} + SP_{(t)} - 1]$$

where $SE_{(t)}$ and $SP_{(t)}$ are, respectively, the **sensitivity** and **specificity** over all possible threshold values *t* of the model. Thus, the **Optimal Model Threshold** corresponds to the model output at the Optimal Cutoff Point.

In the **ROC Table**, GeneXproTools also shows all “SE + SP -1” values and highlights in light green the row with the Optimal Cutoff Point and corresponding Optimal Model Threshold. These parameters are also shown in the Quantiles Statistics Report.



Quantiles: 10 Dataset: Training Set History: Model 1, RS 0.22189084 ☑ Synchronize Tables & Charts

Id	Target	Model Output	True Positives	True Negatives	False Positives	False Negatives	Sensitivity	1-Specificity	SE + SP - 1
4138	1	380.863562819914	2174	1326	326	1174	0.649342891278375	0.197336561743341	0.452006329535034
462	1	380.692084819914	2175	1326	326	1173	0.649641577060932	0.197336561743341	0.452305015317591
498	1	380.692084819914	2176	1326	326	1172	0.649940262843489	0.197336561743341	0.452603701100147
2763	1	380.692084819914	2177	1326	326	1171	0.650238948626045	0.197336561743341	0.452902386882704
3752	1	380.565646	2178	1326	326	1170	0.650537634408602	0.197336561743341	0.453201072665261
3962	0	380.565646	2178	1325	327	1170	0.650537634408602	0.197941888619855	0.452595745788747
2533	1	380.450380819914	2179	1325	327	1169	0.650836320191159	0.197941888619855	0.452894431571304
255	0	379.863562819914	2179	1324	328	1169	0.650836320191159	0.198547215496368	0.452289104694791
4306	1	379.863562819914	2180	1324	328	1168	0.651135005973716	0.198547215496368	0.452587790477348
3517	1	379.738185180086	2181	1324	328	1167	0.651433691756272	0.198547215496368	0.452886476259904

+ **Current Model**

− **General**

Total Cases	5000
Quantiles	10
Positive Cases	3348 (66.96%)
Negative Cases	1652 (33.04%)

+ **Quantiles**

− **ROC**

Area Under the ROC Cur	0.790694672255634
Optimal Cutoff Point (You	0.453201072665261
Optimal Model Threshold	380.565646
Correct	3504 (70.08%)
Wrong	1496 (29.92%)
Total	5000
TP	2178 (43.56%)
TN	1326 (26.52%)
FP	326 (6.52%)
FN	1170 (23.40%)
TPR (Sens)	0.650537634408602
TNR (Spec)	0.802663438256659
FPR (1-Spec)	0.197336561743341

The Optimal Model Threshold can be obviously used to infer a **Confusion Matrix** (in GeneXproTools it is called **ROC Confusion Matrix**) and, in the Cutoff Points Table, you have access to the Predicted Class, the Match, and Type values used to build this ROC Confusion Matrix (you can see the graphical representation of the ROC Confusion Matrix in the Confusion Matrix section).

The visualization of the ROC Confusion Matrix is a valuable tool and can in fact be used to determine the right number of buckets to achieve a good fit with the Logistic Regression Model. But GeneXproTools allows you to do more with the ROC Confusion Matrix and associated Optimal Model Threshold. By allowing the conversion of Logistic Regression runs to the Classification Framework, you can use this model, with its finely adapted Optimal Model Threshold, straightaway to make categorical classifications using the Classification Scoring Engine of GeneXproTools.

The Youden index is also used to evaluate a wide set of useful statistics at the Optimal Cutoff Point (**OCP statistics** for short). They include:

- TP (True Positives)
- TN (True Negatives)
- FP (False Positives)
- FN (False Negatives)
- TPR (True Positives Rate or Sensitivity)
- TNR (True Negatives Rate or Specificity)
- FPR (False Positives Rate, also known as 1-Specificity)
- FNR (False Negatives Rate)
- PPV (Positive Predictive Value)
- NPV (Negative Predictive Value)

- Classification Accuracy (Correct Classifications)
- Classification Error (Wrong Classifications)

How they are calculated is shown in the table below ("TC" represents the number of Total Cases):

TPR (Sensitivity)	TP / (TP + FN)
TNR (Specificity)	TN / (TN + FP)
FPR (1-Specificity)	FP / (FP + TN)
FNR	FN / (FN + TP)
PPV	TP / (TP + FP), and TP + FP ≠ 0
NPV	TN / (TN + FN), and TN + FN ≠ 0
Classification Accuracy	(TP + TN) / TC
Classification Error	(FP + FN) / TC

It is worth pointing out that OCP statistics are quantile-independent and therefore are a good indicator of what could be achieved with a model in terms of logistic fit and accuracy.

◀ | ▶

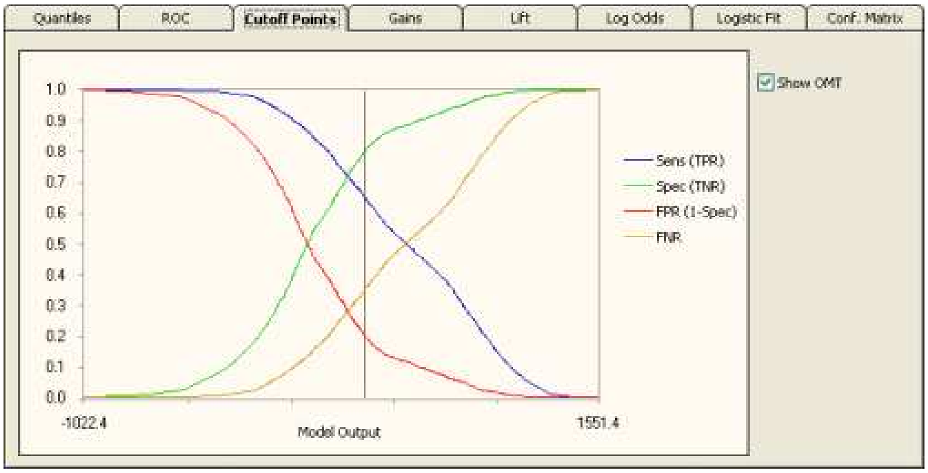
Logistic Regression

Cutoff Points

The **Cutoff Points Analysis** complements nicely the ROC Analysis of the previous section. The **Cutoff Points Chart** shows clearly the intersection of both the sensitivity (TPR) and specificity (TNR) lines and also the intersection of the FPR line with the FNR line. Seeing how these four lines change with model output is a great aid for choosing the **Ideal Cutoff Point** for your test values.

It's true that the Ideal Cutoff Point varies from problem to problem, as one might be interested in minimizing or maximizing different things. Sometimes the goal is to minimize the number of false positives; other times the number false negatives. Other times, though, one might need to maximize the number of true positives or true negatives. With the help of the Cutoff Points Chart of GeneXproTools you can see clearly the best way to move your model threshold to achieve your goals.

Notwithstanding, there is a generic **Optimal Cutoff Point**. This Optimal Cutoff Point is given by the **Youden index** and you can see where it exactly lays in the Cutoff Points Chart. When you tick the Show OMT check box, GeneXproTools draws the **Optimal Model Threshold** in dark brown.

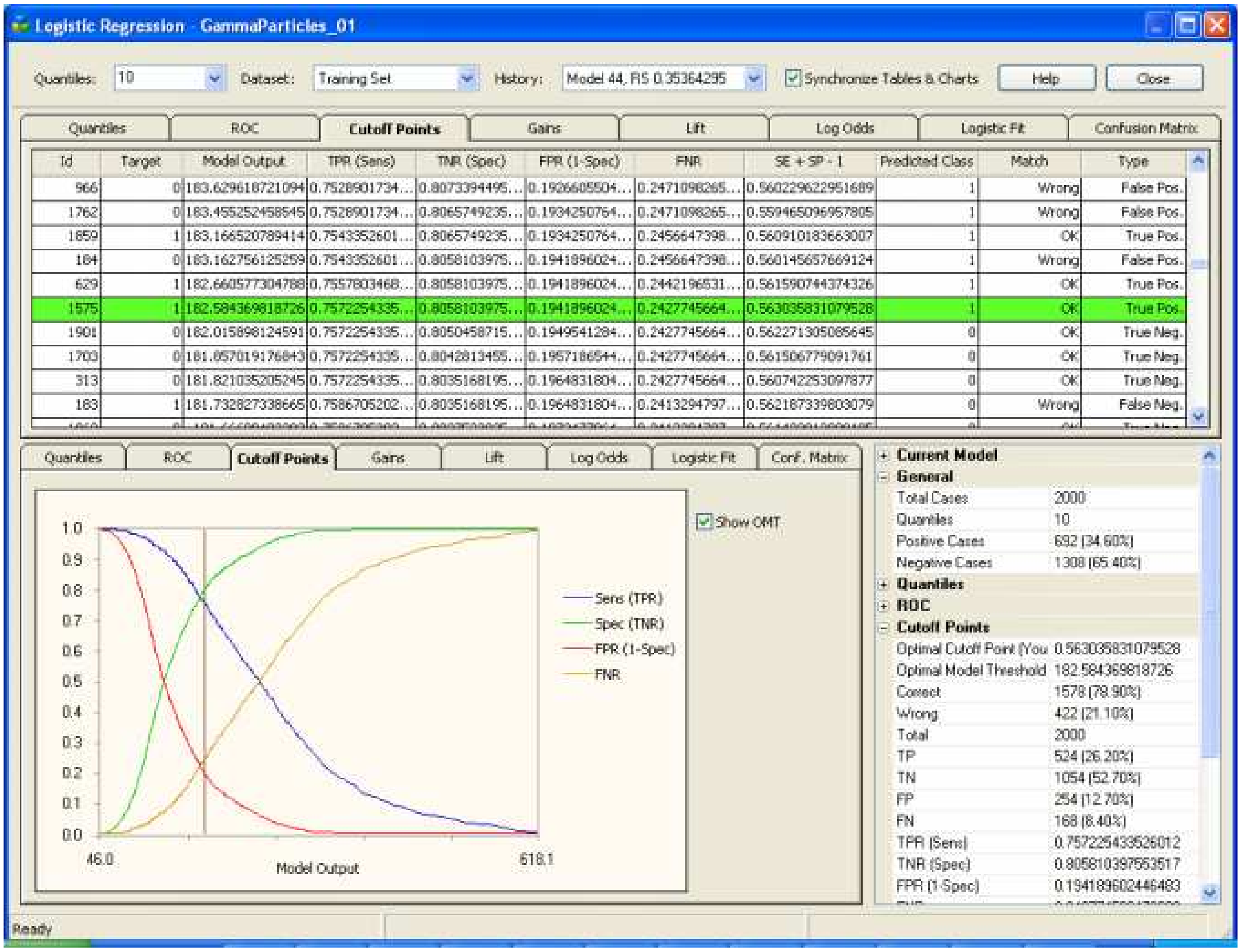


The Youden index *J* returns the maximum value of the expression (for inverted models it returns the minimum):

$$J = \max[SE_{(t)} + SP_{(t)} - 1]$$

where $SE_{(t)}$ and $SP_{(t)}$ are, respectively, the **sensitivity** and **specificity** over all possible threshold values *t* of the model. Thus, the **Optimal Model Threshold** corresponds to the model output at the Optimal Cutoff Point.

In the **Cutoff Points Table**, GeneXproTools also shows all “SE + SP -1” values and highlights in light green the row with the Optimal Cutoff Point and corresponding Optimal Model Threshold. These parameters are also shown in the companion Cutoff Points Statistics Report.



Quantiles: 10 Dataset: Training Set History: Model 44, FIS 0.35364295 ☑ Synchronize Tables & Charts

Id	Target	Model Output	TPR (Sens)	TNR (Spec)	FPR (1-Spec)	FNR	SE + SP - 1	Predicted Class	Match	Type
966	0	183.629618721094	0.7528901734...	0.8073394495...	0.1926605504...	0.2471098265...	0.560229622951689	1	Wrong	False Pos.
1762	0	183.455252458545	0.7528901734...	0.8065749235...	0.1934250764...	0.2471098265...	0.559465096957805	1	Wrong	False Pos.
1859	1	183.166520789414	0.7543352601...	0.8065749235...	0.1934250764...	0.2456647398...	0.560910183663007	1	OK	True Pos.
184	0	183.162756125259	0.7543352601...	0.8058103975...	0.1941896024...	0.2456647398...	0.560145657669124	1	Wrong	False Pos.
629	1	182.660577304788	0.7557803468...	0.8058103975...	0.1941896024...	0.2442196531...	0.561590744374326	1	OK	True Pos.
1575	1	182.584369818726	0.7572254335...	0.8058103975...	0.1941896024...	0.2427745664...	0.563035831079528	1	OK	True Pos.
1901	0	182.015898124591	0.7572254335...	0.8050458715...	0.1949541284...	0.2427745664...	0.562271305085645	0	OK	True Neg.
1703	0	181.857019176843	0.7572254335...	0.8042813455...	0.1957186544...	0.2427745664...	0.561506779091761	0	OK	True Neg.
313	0	181.821035205245	0.7572254335...	0.8035168195...	0.1964831804...	0.2427745664...	0.560742253097877	0	OK	True Neg.
183	1	181.732827338665	0.7586705202...	0.8035168195...	0.1964831804...	0.2413294797...	0.562187339803079	0	Wrong	False Neg.

☑ Show OMT

+ **Current Model**	
− **General**	
Total Cases	2000
Quantiles	10
Positive Cases	692 (34.60%)
Negative Cases	1308 (65.40%)
+ **Quantiles**	
+ **ROC**	
− **Cutoff Points**	
Optimal Cutoff Point (You	0.563035831079528
Optimal Model Threshold	182.584369818726
Correct	1578 (78.90%)
Wrong	422 (21.10%)
Total	2000
TP	524 (26.20%)
TN	1054 (52.70%)
FP	254 (12.70%)
FN	168 (8.40%)
TPR (Sens)	0.757225433526012
TNR (Spec)	0.805810397553517
FPR (1-Spec)	0.194189602446483

The Optimal Model Threshold can be obviously used to infer a **Confusion Matrix** (in GeneXproTools it is called **ROC Confusion Matrix**) and, in the Cutoff Points Table, you have access to the Predicted Class, the Match, and Type values used to build this ROC Confusion Matrix (you can see the graphical representation of the ROC Confusion Matrix in the Confusion Matrix section).

The visualization of the ROC Confusion Matrix is a valuable tool and can in fact be used to determine the right number of buckets to achieve a good fit with the Logistic Regression Model. But GeneXproTools allows you to do more with the ROC Confusion Matrix and associated Optimal Model Threshold. By allowing the conversion of Logistic Regression runs to the Classification Framework, you can use this model, with its finely adapted Optimal Model Threshold, straightaway to make discrete classifications using the Classification Scoring Engine of GeneXproTools.

The Youden index is also used to evaluate a wide set of useful statistics at the Optimal Cutoff Point (**OCP statistics** for short). They include:

- TP (True Positives)
- TN (True Negatives)
- FP (False Positives)
- FN (False Negatives)
- TPR (True Positives Rate or Sensitivity)
- TNR (True Negatives Rate or Specificity)
- FPR (False Positives Rate, also known as 1-Specificity)
- FNR (False Negatives Rate)
- PPV (Positive Predictive Value)
- NPV (Negative Predictive Value)

- Classification Accuracy (Correct Classifications)
- Classification Error (Wrong Classifications)

How they are calculated is shown in the table below ("TC" represents the number of Total Cases):

TPR (Sensitivity)	TP / (TP + FN)
TNR (Specificity)	TN / (TN + FP)
FPR (1-Specificity)	FP / (FP + TN)
FNR	FN / (FN + TP)
PPV	TP / (TP + FP), and TP + FP ≠ 0
NPV	TN / (TN + FN), and TN + FN ≠ 0
Classification Accuracy	(TP + TN) / TC
Classification Error	(FP + FN) / TC

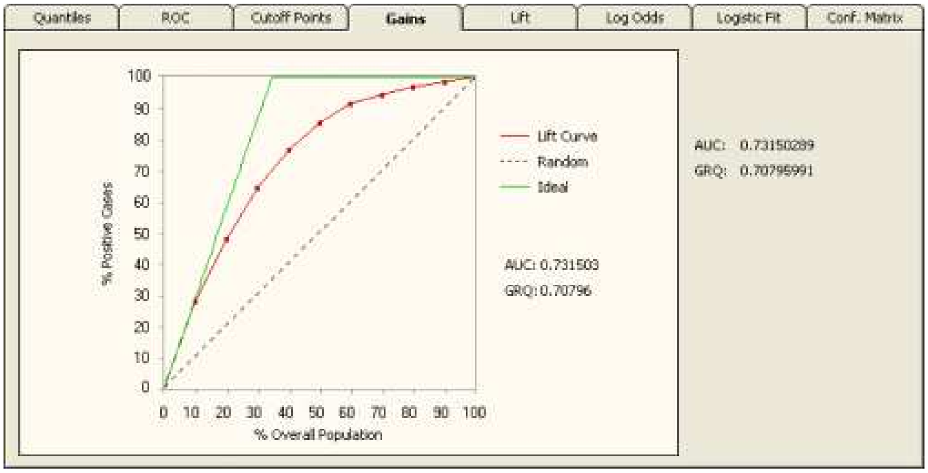
It is worth pointing out that OCP statistics are quantile-independent and therefore are a good indicator of what could be achieved with a model in terms of logistic fit and accuracy.

◀ | ▶

Logistic Regression

Gains Chart

The **Gains Chart** of GeneXproTools is quantile-based and shows the **cumulative gain** as more cases are included in a campaign or test. The **Lift Curve** is compared to both a **Random Model** and an **Ideal Model**, showing clearly the advantages of using a model as opposed to not using one.

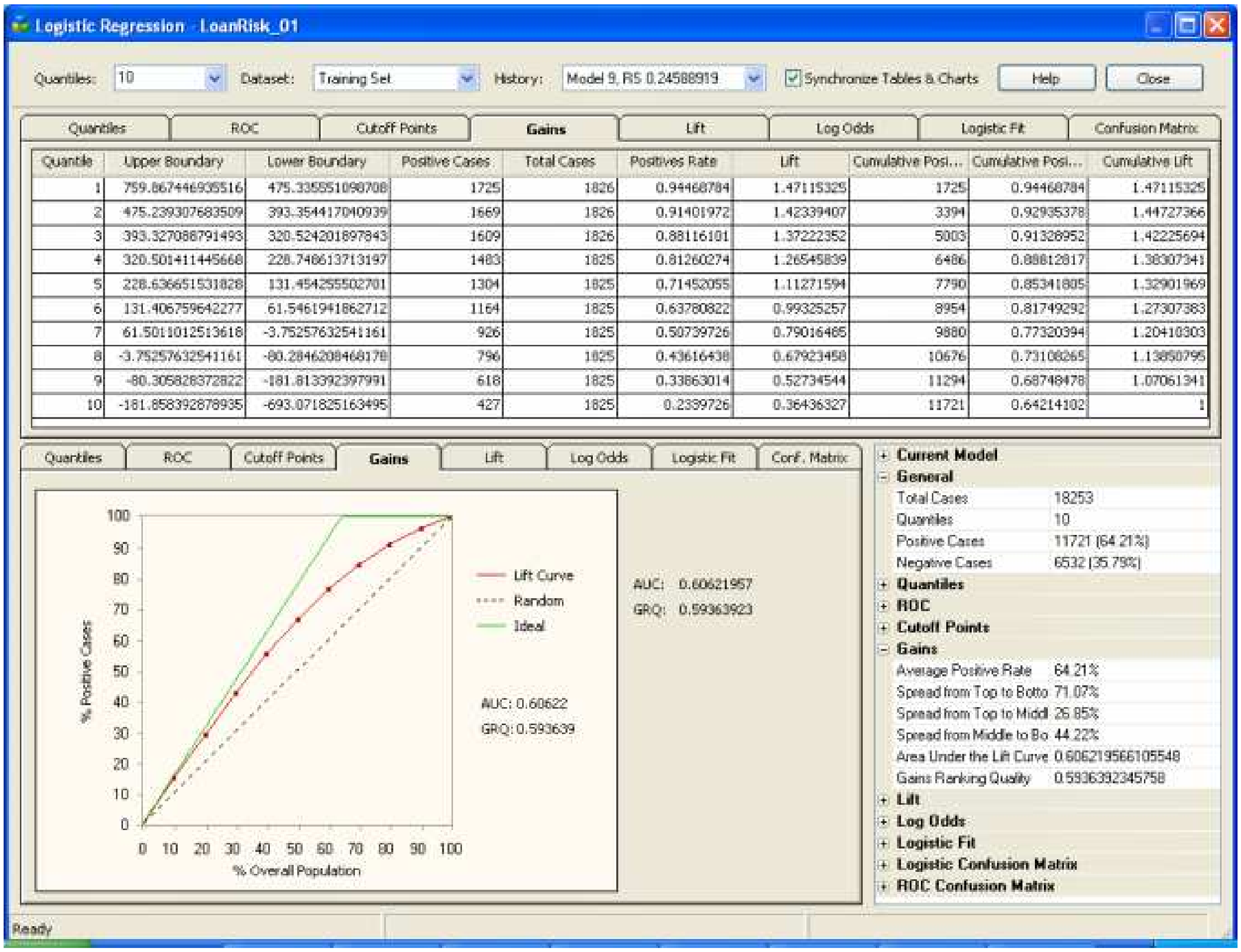


AUC: 0.73150289

GRQ: 0.70795991

The **Random line** in the Gains Chart represents the average response rate. And the **Ideal line** represents a perfect model that is never wrong and therefore could select all the estimated positive responses. So, the further up (or down, for inverted models) the Lift Curve is from the Random Line the better the model.

The **Gains Ranking Quality** (GRQ) is a good indicator of the quality of a model. It is defined as the relation between the area under the Ideal Model and the area under the Lift Curve. It ranges from -1 to +1, with zero corresponding to the Random Model. The better the model the closer the GRQ gets to either +1 or -1 (for inverted perfect models GRQ = -1, whereas for normal perfect models GRQ = 1). As an additional quality measure, the **Area Under the Lift Curve** (represented by **AUC** in the Gains Chart) is also evaluated and shown both in the chart and the companion **Gains Statistics Report**.



Quantiles: 10 Dataset: Training Set History: Model 9, RS 0.24588919 ☑ Synchronize Tables & Charts

Quantile	Upper Boundary	Lower Boundary	Positive Cases	Total Cases	Positives Rate	Lift	Cumulative Posi...	Cumulative Posi...	Cumulative Lift
1	759.867446935516	475.335551090708	1725	1826	0.94468784	1.47115325	1725	0.94468784	1.47115325
2	475.239307683509	393.354417040939	1669	1826	0.91401972	1.42339407	3394	0.92935378	1.44727366
3	393.327088791493	320.524201897843	1609	1826	0.88116101	1.37222352	5003	0.91328952	1.42225694
4	320.501411445668	228.748613713197	1483	1825	0.81260274	1.26545839	6486	0.88812817	1.38307341
5	228.636651531828	131.454255502701	1304	1825	0.71452055	1.11271594	7790	0.85341805	1.32901969
6	131.406759642277	61.5461941862712	1164	1825	0.63780822	0.99325257	8954	0.81749292	1.27307383
7	61.5011012513618	-3.75257632541161	926	1825	0.50739726	0.79016465	9880	0.77320394	1.20410303
8	-3.75257632541161	-80.2846208468178	796	1825	0.43616438	0.67923458	10676	0.73108265	1.13850795
9	-80.305828372822	-181.813392397991	618	1825	0.33863014	0.52734544	11294	0.68748478	1.07061341
10	-181.858392878935	-693.071825163495	427	1825	0.2339726	0.36436327	11721	0.64214102	1

AUC: 0.60622
GRQ: 0.593639

AUC: 0.60621957
GRQ: 0.59363923

- Current Model
- General

Total Cases	18253
Quantiles	10
Positive Cases	11721 (64.21%)
Negative Cases	6532 (35.79%)

- Quantiles
- ROC
- Cutoff Points
- Gains

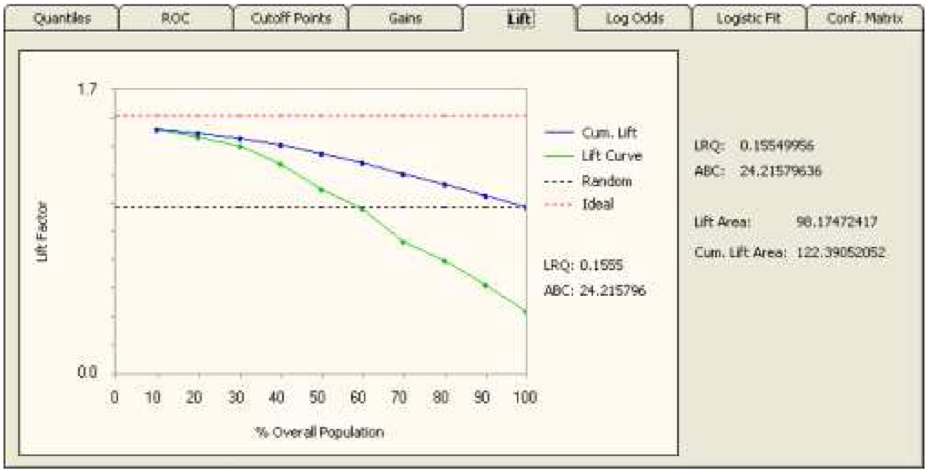
Average Positive Rate	64.21%
Spread from Top to Botto	71.07%
Spread from Top to Middl	26.85%
Spread from Middle to Bo	44.22%
Area Under the Lift Curve	0.606219566105548
Gains Ranking Quality	0.593639234575 8

- Lift
- Log Odds
- Logistic Fit
- Logistic Confusion Matrix
- ROC Confusion Matrix

Logistic Regression

Lift Chart

The **Lift Chart** of GeneXproTools shows both the **Lift Curve** and **Cumulative Lift Curve** on the same graph. These curves are also shown in relation to a **Random Model** and an **Ideal Model**.



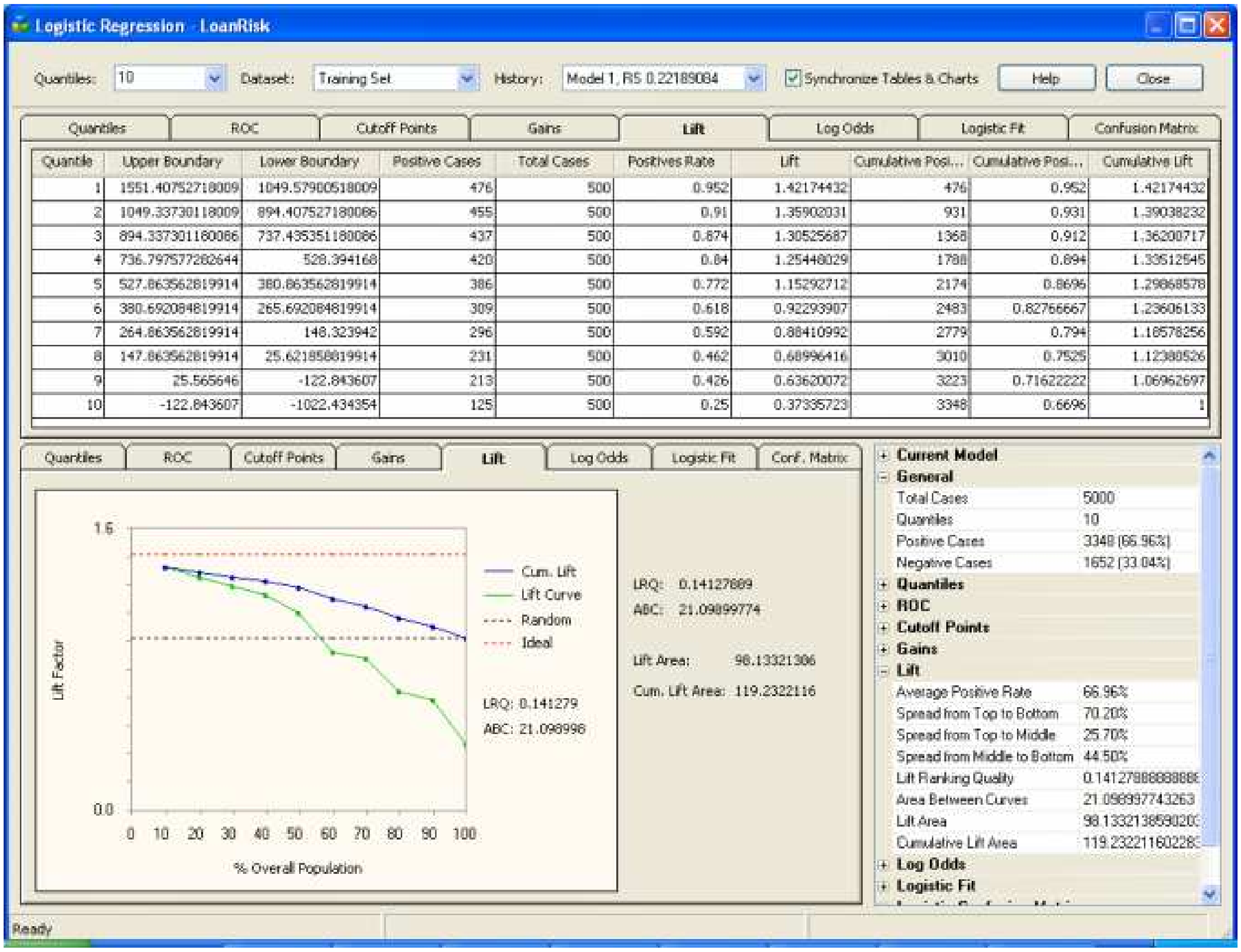
LRQ: 0.15549956
ABC: 24.21579636

Lift Area: 98.17472417
Cum. Lift Area: 122.39052052

The **Random line** in the Lift Chart represents the average response rate. And the **Ideal line** represents a perfect model that is never wrong and therefore could select all the estimated positive responses. The point where the Lift Curve crosses the Random line corresponds approximately to the percentage of the population beyond which the benefits from using the model are lost.

Other useful visual clues from the Lift Chart include the **Area Between both Lift Curves** (represented by **ABC** in the Lift Chart). Theoretically, the greater ABC the better the model. The individual areas under each of the Lift Curves are also computed and shown both on the Lift Chart and in the companion **Lift Statistics Report**.

The **Lift Ranking Quality** (LRQ) is yet another useful indicator of the accuracy of a model. It corresponds to the ABC area normalized against the area under the Ideal line. Negative values both for the ABC and LRQ are indicative of an inverted model.


Logistic Regression - LoanRisk
Quantiles | ROC | Cutoff Points | Gains | Lift | Log Odds | Logistic Fit | Confusion Matrix
Quantiles | ROC | Cutoff Points | Gains | Lift | Log Odds | Logistic Fit | Conf. Matrix
1.6
0.0
Lift Factor
0
10
20
30
40
50
60
70
80
90
100
% Overall Population
Cum. Lift
Lift Curve
Random
Ideal
Help
Close
Ready

Quantiles: 10 Dataset: Training Set History: Model 1, R5 0.22189084 ☑ Synchronize Tables & Charts

Quantile	Upper Boundary	Lower Boundary	Positive Cases	Total Cases	Positives Rate	Lift	Cumulative Posi...	Cumulative Posi...	Cumulative Lift
1	1551.40752718009	1049.57900518009	476	500	0.952	1.42174432	476	0.952	1.42174432
2	1049.33730118009	894.407527180086	455	500	0.91	1.35902031	931	0.931	1.39038232
3	894.337301180086	737.435351180086	437	500	0.874	1.30525687	1368	0.912	1.36200717
4	736.797577282644	528.394168	420	500	0.84	1.25448029	1788	0.894	1.33512545
5	527.863562819914	380.863562819914	386	500	0.772	1.15292712	2174	0.8696	1.29868578
6	380.692084819914	265.692084819914	309	500	0.618	0.92293907	2483	0.82766667	1.23606133
7	264.863562819914	148.323942	296	500	0.592	0.88410992	2779	0.794	1.18578256
8	147.863562819914	25.621858819914	231	500	0.462	0.68996416	3010	0.7525	1.12380526
9	25.565646	-122.843607	213	500	0.426	0.63620072	3223	0.71622222	1.06962697
10	-122.843607	-1022.434354	125	500	0.25	0.37335723	3348	0.6696	1

LRQ: 0.141279
ABC: 21.098998

LRQ: 0.14127889
ABC: 21.09899774

Lift Area: 98.13321306
Cum. Lift Area: 119.2322116

+ **Current Model**
− **General**

Total Cases	5000
Quantiles	10
Positive Cases	3348 (66.96%)
Negative Cases	1652 (33.04%)

+ **Quantiles**
+ **ROC**
+ **Cutoff Points**
+ **Gains**
− **Lift**

Average Positive Rate	66.96%
Spread from Top to Bottom	70.20%
Spread from Top to Middle	25.70%
Spread from Middle to Bottom	44.50%
Lift Ranking Quality	0.141278888888
Area Between Curves	21.098997743263
Lift Area	98.1332138590203
Cumulative Lift Area	119.232211602283

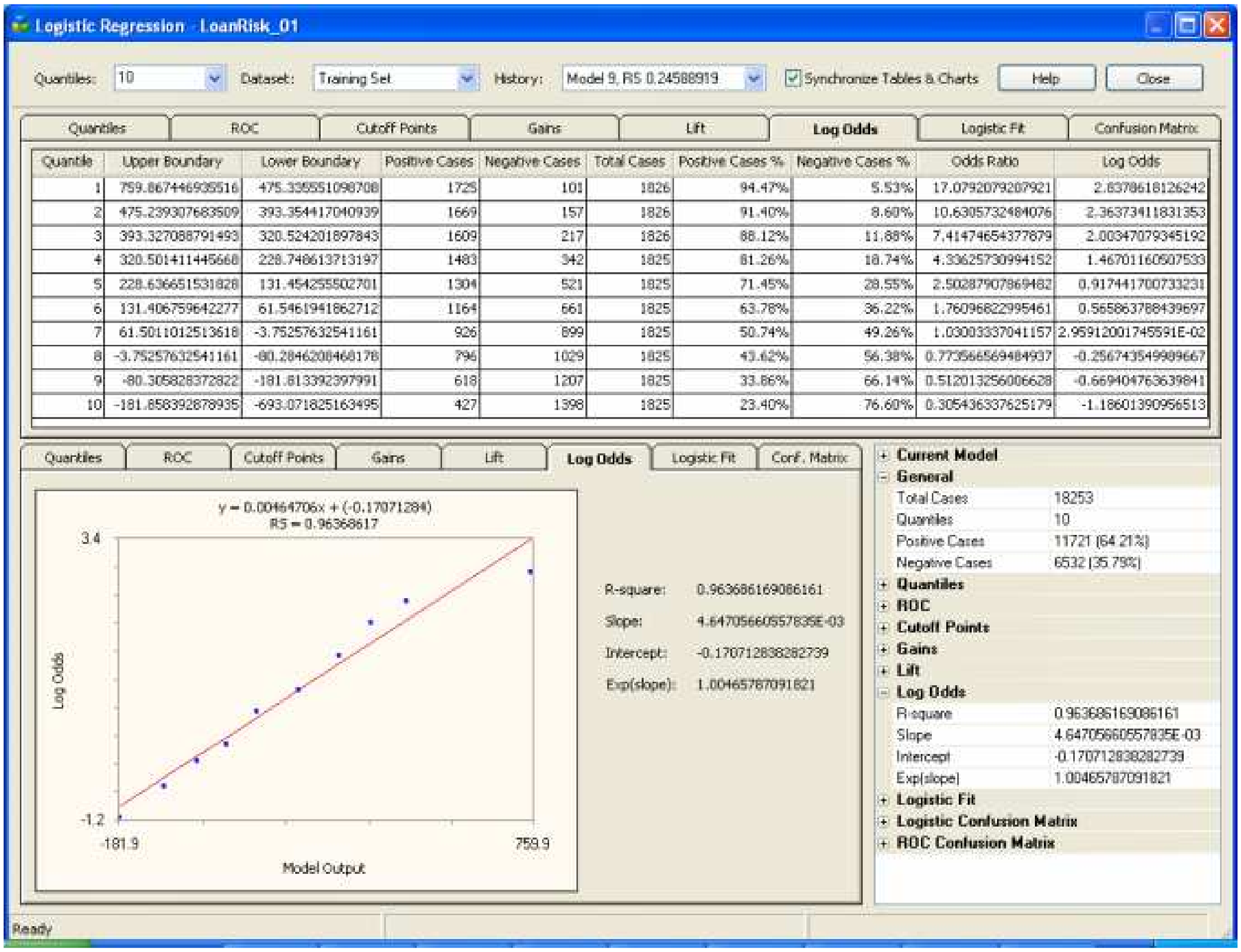
+ **Log Odds**
+ **Logistic Fit**

Logistic Regression

Log Odds and Logistic Regression

The **Log Odds Chart** is vital to the **Logistic Regression Model**. It's with its aid that the **slope** and **intercept** of the Logistic Regression Model are calculated. And the procedure is quite simple. As mentioned previously, it's quantile-based and, in fact, just a few additional calculations are required to draw the regression line.

So, based on the Quantile Table, one first evaluates the **odds ratio** for all the buckets (you can check all the values on the **Log Odds Table** under Odds Ratio). Then the natural logarithm of this ratio is evaluated, resulting in what goes by the name of **Log Odds** (the log odds values are also shown on the Log Odds Table under Log Odds).



Quantiles: 10 Dataset: Training Set History: Model 9, RS 0.24588919 ☑ Synchronize Tables & Charts

Quantile	Upper Boundary	Lower Boundary	Positive Cases	Negative Cases	Total Cases	Positive Cases %	Negative Cases %	Odds Ratio	Log Odds
1	759.867446935516	475.335551098708	1725	101	1826	94.47%	5.53%	17.0792079207921	2.8378618126242
2	475.239307683509	393.354417040939	1669	157	1826	91.40%	8.60%	10.6305732484076	2.36373411831353
3	393.327088791493	320.524201897843	1609	217	1826	88.12%	11.88%	7.41474654377879	2.00347079345192
4	320.501411445668	228.748613713197	1483	342	1825	81.26%	18.74%	4.33625730994152	1.46701160507533
5	228.636651531828	131.454255502701	1304	521	1825	71.45%	28.55%	2.50287907869482	0.917441700733231
6	131.406759642277	61.5461941862712	1164	661	1825	63.78%	36.22%	1.76096822995461	0.565863788439697
7	61.5011012513618	-3.75257632541161	926	899	1825	50.74%	49.26%	1.03003337041157	2.95912001745591E-02
8	-3.75257632541161	-80.2846208468178	796	1029	1825	43.62%	56.38%	0.773566569484937	-0.256743549989667
9	-80.305828372822	-181.813392397991	618	1207	1825	33.86%	66.14%	0.512013256006628	-0.669404763639841
10	-181.858392878935	-693.071825163495	427	1398	1825	23.40%	76.60%	0.305436337625179	-1.18601390956513

R-square: 0.963686169086161
Slope: 4.64705660557835E-03
Intercept: -0.170712838282739
Exp(slope): 1.00465787091821

+ Current Model
− General

Total Cases	18253
Quantiles	10
Positive Cases	11721 (64.21%)
Negative Cases	6532 (35.79%)

+ Quantiles
+ ROC
+ Cutoff Points
+ Gains
+ Lift
− Log Odds

R-square	0.963686169086161
Slope	4.64705660557835E-03
Intercept	-0.170712838282739
Exp(slope)	1.00465787091821

+ Logistic Fit
+ Logistic Confusion Matrix
+ ROC Confusion Matrix

Note, however, that there might be a problem in the evaluation of the log odds if we have buckets with zero positive cases. But this problem can be easily fixed. Although rare for large datasets, it can sometimes happen that some of the buckets end up with zero positive cases in them. And this obviously results in a calculation error in the evaluation of the natural logarithm of the odds ratio. GeneXproTools handles this with a slight modification to the **Laplace estimator** to get what is called a complete **Bayesian formulation** with prior probabilities. In essence, this means that if the quantile table we are using has buckets with no positive cases in them, then we do the equivalent of priming all the buckets with a very small amount of positive cases.

The formula GeneXproTools uses in the evaluation of the Positives Rate values p_i for all the quantiles is the following:

$$p_i = \frac{Q_i + \mu \cdot P}{T_i + \mu}$$

where μ is the Laplace estimator that in GeneXproTools has the value of 0.01; Q_i and T_i are, respectively, the number of Positive Cases and the number of Total Cases in bucket *i*; and *P* is the Average Positive Rate of the whole dataset.

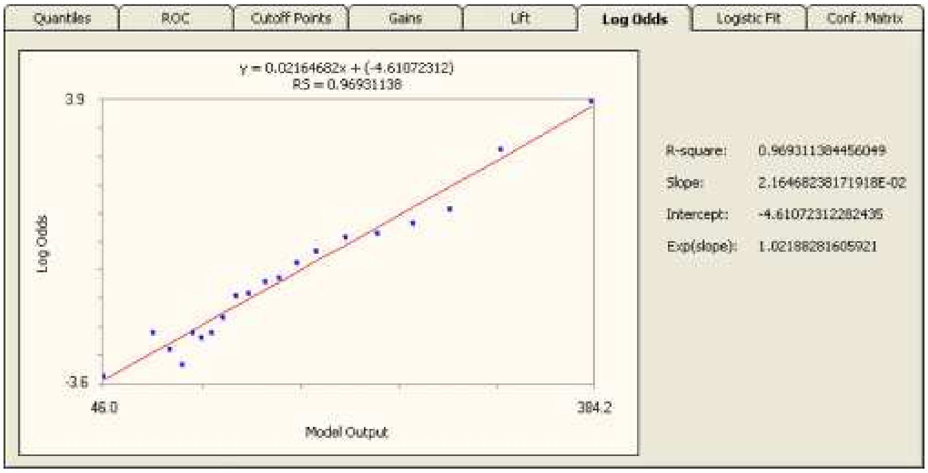
So, in the **Log Odds Chart**, the Log Odds values (adjusted or not with the Laplace strategy) are plotted on the Y-axis against the Model Output in the X-axis. And as for Quantile Regression, here there are also special rules to follow, depending on whether the predominant class is “1” or “0” and whether the model is normal or inverted. To be precise, the Log Odds are plotted against the Model Upper Boundaries if the predominant class is “1” and the model is normal, or the predominant class is “0” and the model is inverted; or against the Lower Boundaries if the predominant class is “1” and the model is inverted, or the predominant class is “0” and the model is normal.

Then a **weighted linear regression** is performed and the **slope** and **intercept** of the regression line are evaluated. And these are the parameters that will be used in the **Logistic Regression Equation** to evaluate the probabilities.

The regression line can be written as:

$$\log\left(\frac{p}{1-p}\right) = ax + b$$

where *p* is the probability of being “1”; *x* is the Model Output; and *a* and *b* are, respectively, the slope and intercept of the regression line. GeneXproTools draws the regression line and shows both the equation and the R-square in the **Log Odds Chart**.



R-square: 0.969311384456049

Slope: 2.16468238171918E-02

Intercept: -4.61072312282435

Exp(slope): 1.02188281605921

And now solving the logistic equation above for *p*, gives:

$$p = \frac{1}{1+e^{-(ax+b)}}$$

which is the formula for evaluating the **probabilities** with the Logistic Regression Model. The probabilities estimated for each are shown in the Logistic Fit Table.

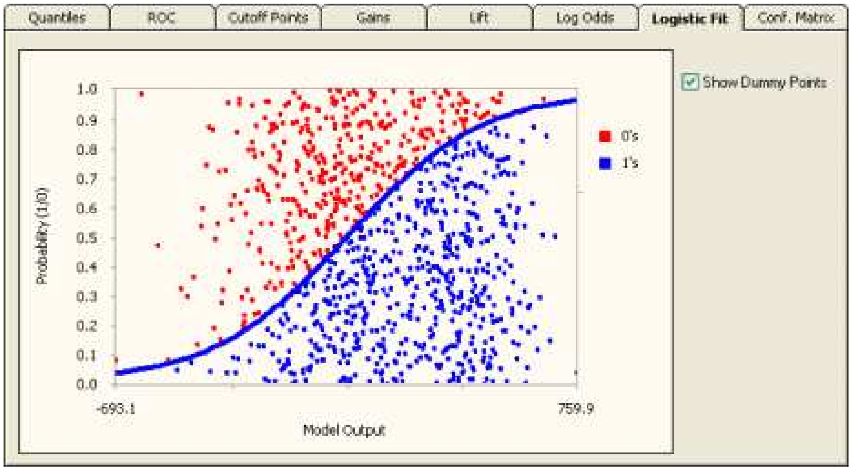
Besides the slope and intercept of the Logistic Regression Model, another useful and popular parameter is the exponent of the slope, usually represented by **Exp(slope)**. It describes the proportionate rate at which the predicted odds ratio changes with each successive unit of *x*. GeneXproTools also shows this parameter both in the Log Odds Chart and in the companion **Log Odds Stats Report**.

◀ | ▶

Logistic regression

Logistic Fit Chart

The **Logistic Fit Chart** is a very useful graph that allows not only a quick visualization of how good the **Logistic Fit** is (the shape and steepness of the **sigmoid curve** are excellent indicators of the accuracy of the model), but also how the model outputs are distributed all over the model range.



The blue line (the sigmoid curve) on the graph is the **logistic transformation** of the model outputs *x*, using the **slope *a*** and **intercept *b*** calculated in the Log Odds Chart and is evaluated by the already familiar formula for the probability *p*:

$$p = \frac{1}{1+e^{-(ax+b)}}$$

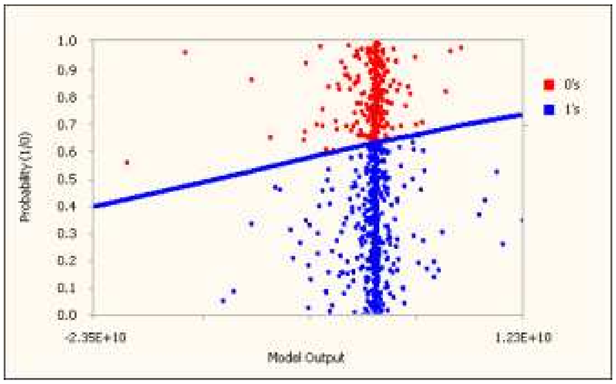
Since the proportion of Positive (1's) responses and Negative (0's) responses must add up to 1, both probabilities can be read on the vertical axis on the left. Thus, the **probability of "1"** is read directly on the vertical axis; and the **probability of "0"** is the distance from the line to the top of the graph, which is 1 minus the axis reading.

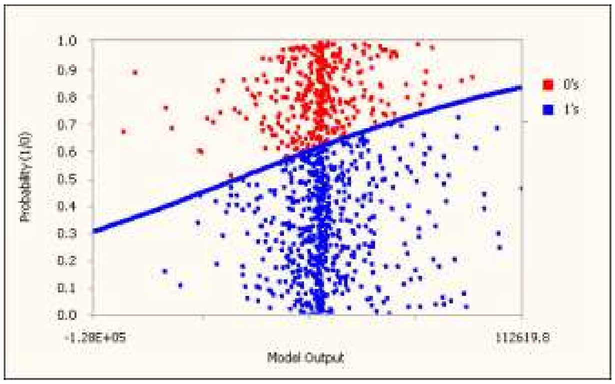
But there's more information on the Logistic Fit Chart. Firstly, the vertical axis on the right shows the proportion of Positive and Negative cases in the dataset. Then, by plotting the **dummy data points**, which consist of up to 1000 randomly selected model scores paired with dummy random ordinates, one can clearly visualize how model scores are dispersed. Are they all clumped together or are they finely distributed, which is the telltale sign of a good model? This is valuable information not only to guide the modeling process (not only in choosing model architecture and composition but also in the exploration of different fitness functions and class encodings that you can use to model your data), but also to sharpen one's intuition and knowledge about the workings of learning evolutionary systems.

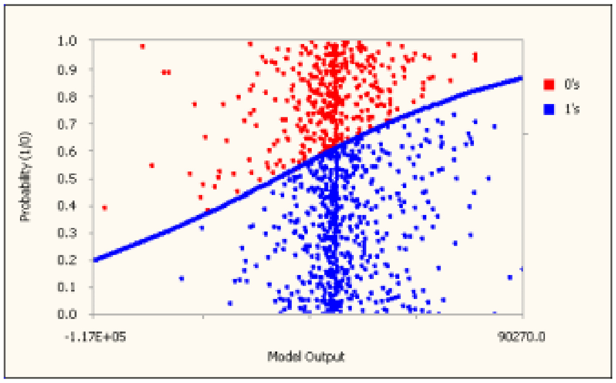
Indeed, browsing through the different models created in a run might prove both insightful and great fun. And you can do that easily as all the models in the Run History are accessible through the History combo box in the Logistic Regression Window. Good models will generally allow for a good distribution, resulting in a unique score for each different case. Bad models, though, will usually concentrate most of their responses around certain values and consequently are unable to distinguish between most cases.

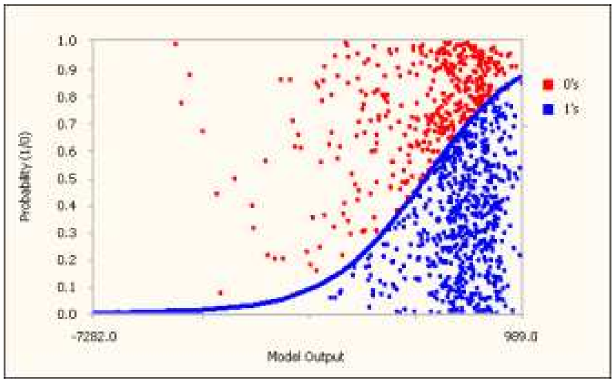
Below is shown a **Gallery of Logistic Fit Charts** typical of intermediate models generated during a GeneXproTools run. It was generated using the same models used to create the twin ROC Curve Gallery presented in the ROC Analysis section. The models were created for a risk assessment problem with a training dataset with 18,253 cases and using a small population of just 30 programs. Both the R-square and the Area Under the ROC Curve (AUC) of each model, as well as the generation at which they were discovered, are also shown as reference. From left to right and top to bottom, they correspond to the following:

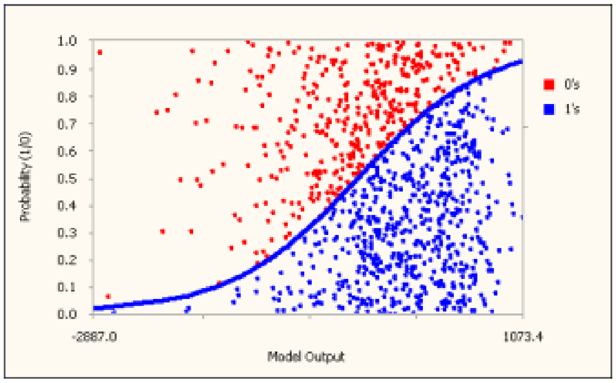
- Generation 0, R-square = 0.002221, AUC = 0.544463
- Generation 3, R-square = 0.022389, AUC = 0.584494
- Generation 8, R-square = 0.050686, AUC = 0.635251
- Generation 12, R-square = 0.064736, AUC = 0.696237
- Generation 14, R-square = 0.163695, AUC = 0.746642
- Generation 344, R-square = 0.219212, AUC = 0.782164

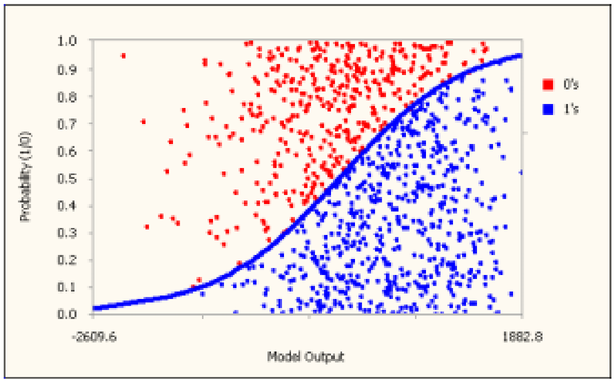












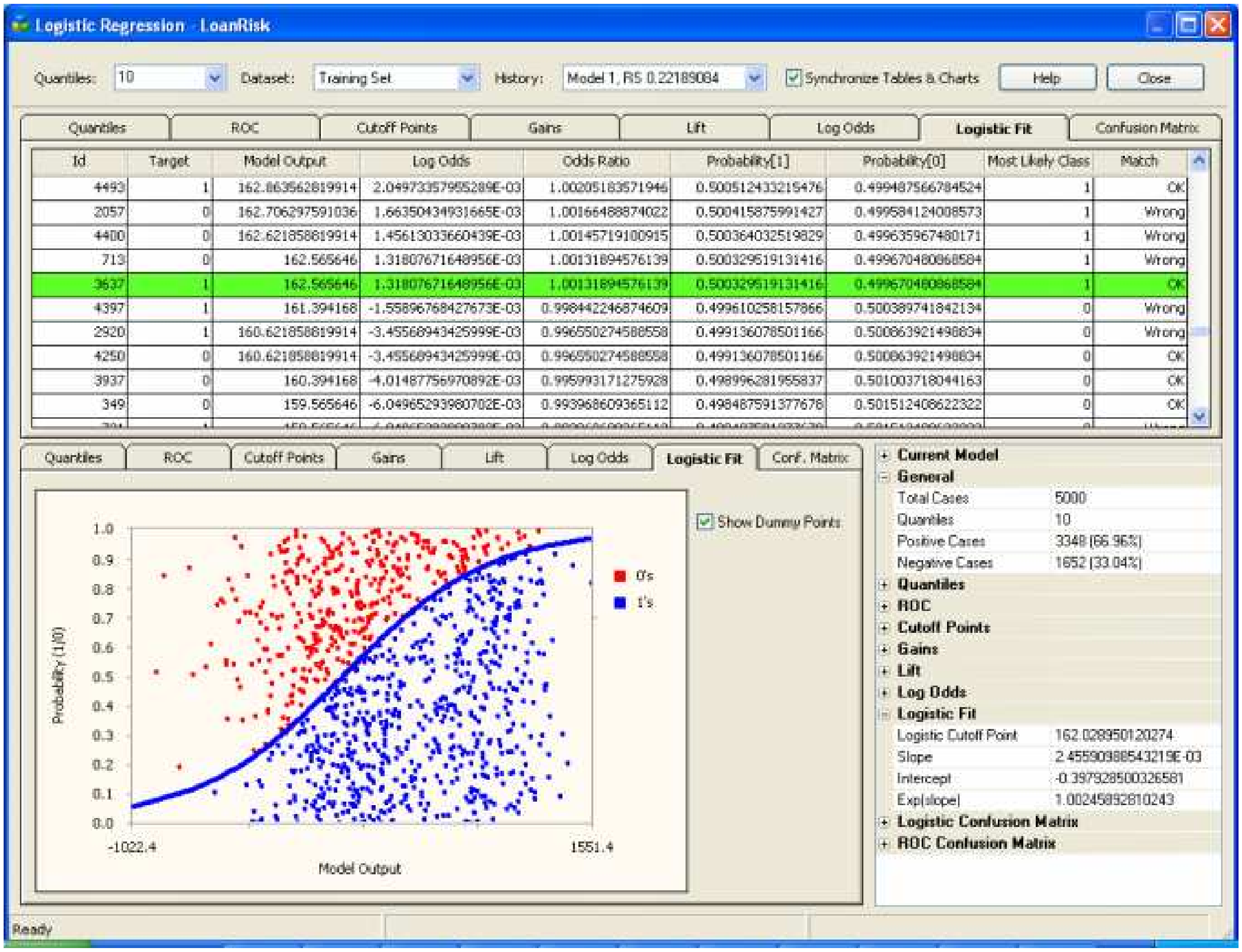
Besides its main goal, which is to estimate the probability of a response, the Logistic Regression Model can also be used to make **categorical predictions**. From the logistic regression equation introduced in the previous section, we know that when a Positive event has the same probability of happening as a Negative one, the log odds term in the logistic regression equation becomes zero, giving:

$$x = -\frac{b}{a}$$

where *x* is the model output at the **Logistic Cutoff Point**; and *a* and *b* are, respectively, the slope and the intercept of the regression line.

The Logistic Cutoff Point can be obviously used to infer a **Confusion Matrix** (in GeneXproTools it is called **Logistic Confusion Matrix**), with model scores resulting in probabilities(1) higher than or equal to 0.5 being converted into Positive cases or into Negative otherwise.

In the **Logistic Fit Table**, GeneXproTools shows the Most Likely Class, the Match, and Type values used to build this Logistic Confusion Matrix (you can see the graphical representation of the Logistic Confusion Matrix in the Confusion Matrix Tab). For easy visualization, the model output closest to the Logistic Cutoff Point is highlighted in light green in the Logistic Fit Table. Note that the exact value of the Logistic Cutoff Point is shown in the companion **Logistic Fit Stats Report**.



Logistic Regression - LoanRisk

Quantiles: 10 Dataset: Training Set History: Model 1, RS 0.22189084 ☑ Synchronize Tables & Charts

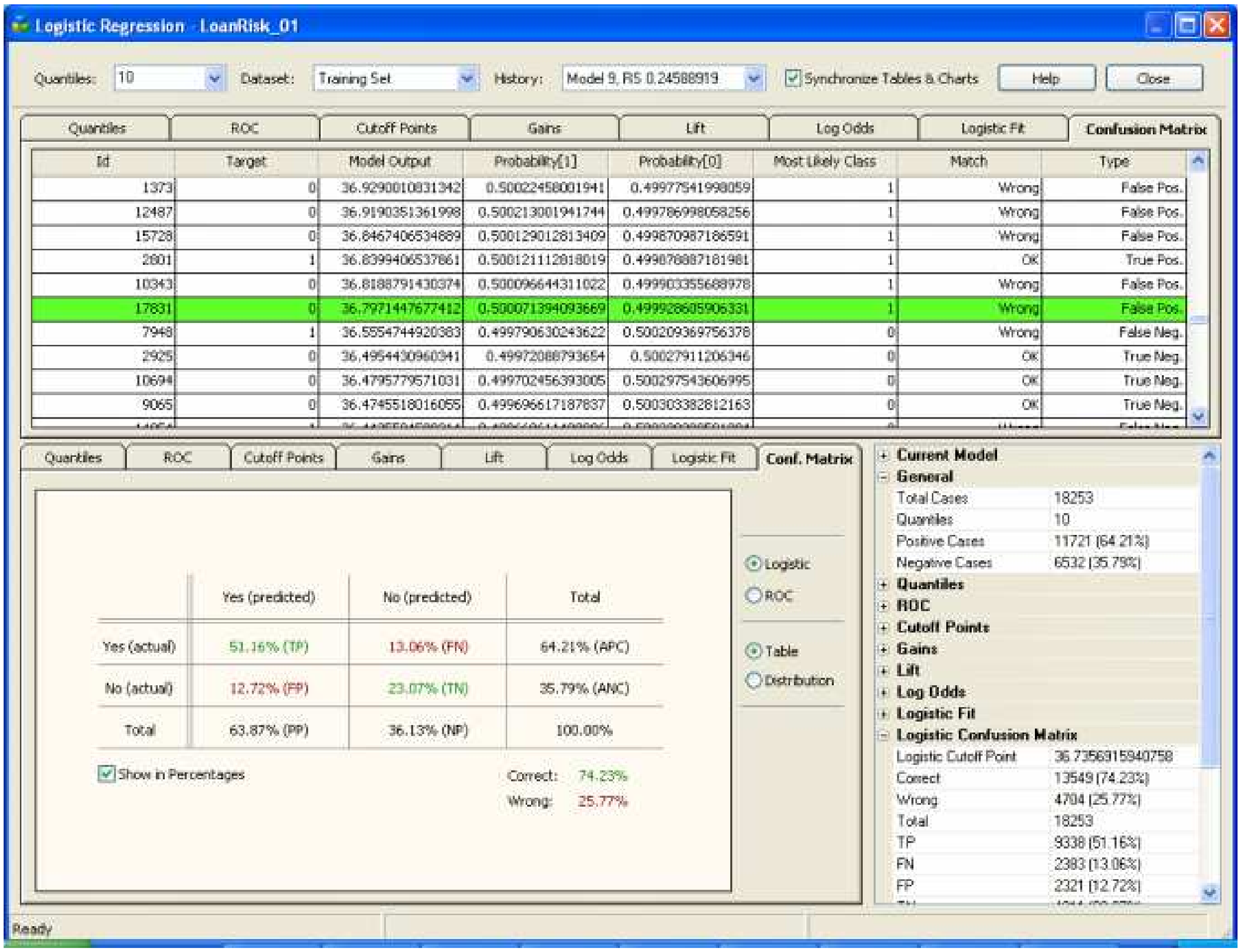
Id	Target	Model Output	Log Odds	Odds Ratio	Probability[1]	Probability[0]	Most Likely Class	Match
4493	1	162.863562819914	2.04973357955289E-03	1.00205183571946	0.500512433215476	0.499487566784524	1	OK
2057	0	162.706297591036	1.66350434931665E-03	1.00166488874022	0.500415875991427	0.499584124008573	1	Wrong
4400	0	162.621858819914	1.45613033660439E-03	1.00145719100915	0.500364032519829	0.499635967480171	1	Wrong
713	0	162.565646	1.31807671648956E-03	1.00131894576139	0.500329519131416	0.499670480868584	1	Wrong
3637	1	162.565646	1.31807671648956E-03	1.00131894576139	0.500329519131416	0.499670480868584	1	OK
4397	1	161.394168	-1.55896768427673E-03	0.998442246874609	0.499610258157866	0.500389741842134	0	Wrong
2920	1	160.621858819914	-3.45568943425999E-03	0.996550274588558	0.499136078501166	0.500863921498834	0	Wrong
4250	0	160.621858819914	-3.45568943425999E-03	0.996550274588558	0.499136078501166	0.500863921498834	0	OK
3937	0	160.394168	-4.01487756970892E-03	0.995993171275928	0.498996281955837	0.501003718044163	0	OK
349	0	159.565646	-6.04965293980702E-03	0.993968609365112	0.498487591377678	0.501512408622322	0	OK

- Current Model
- General
  - Total Cases 5000
  - Quantiles 10
  - Positive Cases 3348 (66.96%)
  - Negative Cases 1652 (33.04%)
- Quantiles
- ROC
- Cutoff Points
- Gains
- Lift
- Log Odds
- Logistic Fit
  - Logistic Cutoff Point 162.028950120274
  - Slope 2.45590988543219E-03
  - Intercept -0.397928500326581
  - Exp(slope) 1.00245892810243
- Logistic Confusion Matrix
- ROC Confusion Matrix

Logistic Regression

Confusion Matrix

In its Logistic Regression Framework, GeneXproTools infers and shows two different **Confusion Matrixes**: the **Logistic Confusion Matrix** and the **ROC Confusion Matrix**. Both these matrixes are excellent indicators of the accuracy of a model (of both the core model and the final logistic regression model), but they can also be used to fine-tune the logistic regression model.



Quantiles: 10 Dataset: Training Set History: Model 9, RS 0.24588919 ☑ Synchronize Tables & Charts

Id	Target	Model Output	Probability[1]	Probability[0]	Most Likely Class	Match	Type
1373	0	36.9290010831342	0.50022458001941	0.49977541998059	1	Wrong	False Pos.
12487	0	36.9190351361998	0.500213001941744	0.499786998058256	1	Wrong	False Pos.
15728	0	36.8467406534889	0.500129012813409	0.499870987186591	1	Wrong	False Pos.
2801	1	36.8399406537861	0.500121112818019	0.499878887181981	1	OK	True Pos.
10343	0	36.8188791430374	0.500096644311022	0.499903355688978	1	Wrong	False Pos.
17831	0	36.7971447677412	0.500071394093669	0.499928605906331	1	Wrong	False Pos.
7948	1	36.5554744920383	0.499790630243622	0.500209369756378	0	Wrong	False Neg.
2925	0	36.4954430960341	0.49972088793654	0.50027911206346	0	OK	True Neg.
10694	0	36.4795779571031	0.499702456393005	0.500297543606995	0	OK	True Neg.
9065	0	36.4745518016055	0.499696617187837	0.500303382812163	0	OK	True Neg.

	Yes (predicted)	No (predicted)	Total
Yes (actual)	51.16% (TP)	13.06% (FN)	64.21% (APC)
No (actual)	12.72% (FP)	23.07% (TN)	35.79% (ANC)
Total	63.87% (PP)	36.13% (NP)	100.00%

☑ Show in Percentages

Correct: 74.23%
Wrong: 25.77%

+ **Current Model**	
− **General**	
Total Cases	18253
Quantiles	10
Positive Cases	11721 (64.21%)
Negative Cases	6532 (35.79%)
+ **Quantiles**	
+ **ROC**	
+ **Cutoff Points**	
+ **Gains**	
+ **Lift**	
+ **Log Odds**	
+ **Logistic Fit**	
− **Logistic Confusion Matrix**	
Logistic Cutoff Point	36.7356915940758
Correct	13549 (74.23%)
Wrong	4704 (25.77%)
Total	18253
TP	9338 (51.16%)
FN	2383 (13.06%)
FP	2321 (12.72%)

The **Logistic Confusion Matrix** is derived from the logistic regression model and infers the Most Likely Class through the predicted probabilities evaluated for each sample case. Thus, probabilities higher than or equal to 0.5 (the **Logistic Cutoff Point**) indicate a Positive response or a Negative otherwise. The model output closest to the Logistic Cutoff Point is highlighted in light green in the Confusion Matrix Table. Note that the exact value of the Logistic Cutoff Point is shown in the companion **Logistic Confusion Matrix Stats Report**.

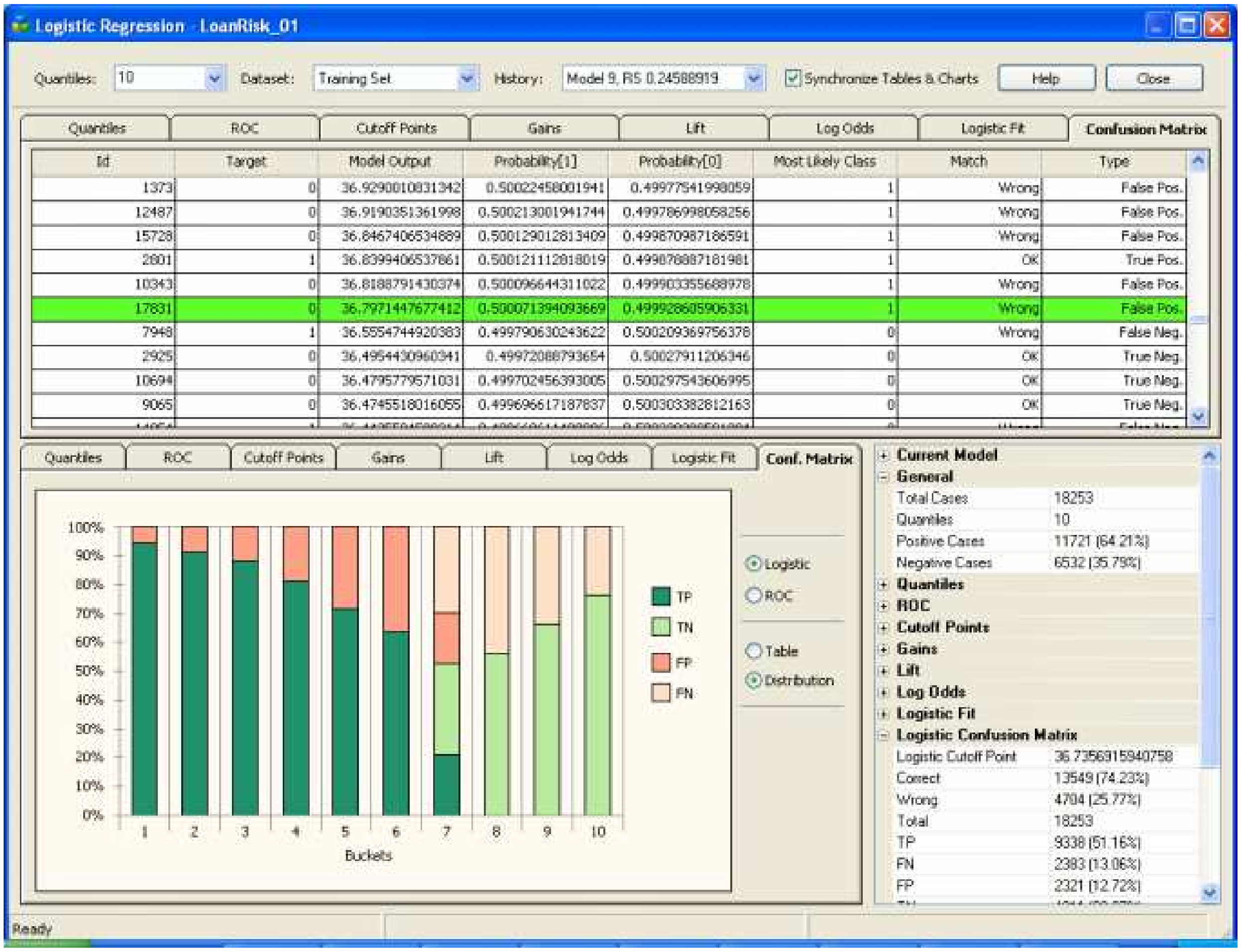
In the **Confusion Matrix Table** you have access not only to the predicted probabilities for each class but also to the Most Likely Class plus how these predictions compare to actual target values. GeneXproTools also shows in the table the Type of each classification (**true positive**, **true negative**, **false positive**, or **false negative**) for all sample cases, which are obviously all the calculations you need to build the Confusion Matrix that is displayed in the graphic section.

The **ROC Confusion Matrix**, on the other hand, is inferred using the Optimal Cutoff Point, a parameter derived from the ROC Curve. This means that for model scores higher than or equal to the Optimal Model Threshold, a Positive response is predicted; and a Negative response otherwise. Note that, despite displaying here in this section the diagram representation of the ROC Confusion Matrix, the confusion matrix data (Predicted Class, Match, and Type) are shown in the Cutoff Points Table.

Note, however, that the statistics evaluated at the Optimal Cutoff Point (or **OCP statistics**, for short) might result in slightly different values than the ones derived from the ROC Confusion Matrix. Remember that OCP statistics are evaluated using the direct readings of all the parameters at the Optimal Cutoff

Point (this point, which is highlighted in green both in the ROC Curve Table and Cutoff Points Table, is also highlighted here in green for a comparison with the Logistic Cutoff Point). For inverted models, for instance, the ROC Confusion Matrix was adjusted to match the default predictions of binomial logistic regression, which always predicts the "1" or positive class. The OCP statistics, however, are not adjusted for inversion and correspond to the actual values for the model. Also note that if you decide to export an inverted model to the Classification Framework, the confusion matrix you'll get there using the Optimal Model Threshold will match the OCP statistics rather than the ROC Confusion Matrix.

Besides the canonical confusion matrix, GeneXproTools also shows a **new kind of confusion matrix**. This new confusion matrix plots the **distribution** of all the classification outcomes (TP, TN, FP, FN) along the different quantiles or buckets. This shows clearly what each model is doing, and where their strengths and weaknesses lay. And by comparing both **Distribution Confusion Matrixes** (logistic and ROC), you can also see how both systems are operating. This is valuable information that you can use in different ways, but most importantly you can use it immediately to fine-tune the number of quantiles in your system so that you can get the most of the logistic fit (as a reminder, the ROC Confusion Matrix is quantile-independent and can be used as reference for fine-tuning the logistic regression model that is quantile dependent).


Logistic Regression - LoanRisk_01
Quantiles | ROC | Cutoff Points | Gains | Lift | Log Odds | Logistic Fit | Confusion Matrix
Quantiles | ROC | Cutoff Points | Gains | Lift | Log Odds | Logistic Fit | Conf. Matrix
Help
Close
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
6
7
8
9
10
Buckets
TP
TN
FP
FN
Ready

Quantiles: 10 Dataset: Training Set History: Model 9, RS 0.24588919 ☑ Synchronize Tables & Charts

Id	Target	Model Output	Probability[1]	Probability[0]	Most Likely Class	Match	Type
1373	0	36.9290010831342	0.50022458001941	0.49977541998059	1	Wrong	False Pos.
12487	0	36.9190351361998	0.500213001941744	0.499786998058256	1	Wrong	False Pos.
15728	0	36.8467406534889	0.500129012813409	0.499870987186591	1	Wrong	False Pos.
2801	1	36.8399406537861	0.500121112818019	0.499878887181981	1	OK	True Pos.
10343	0	36.8188791430374	0.500096644311022	0.499903355688978	1	Wrong	False Pos.
17831	0	36.7971447677412	0.500071394093669	0.499928605906331	1	Wrong	False Pos.
7948	1	36.5554744920383	0.499790630243622	0.500209369756378	0	Wrong	False Neg.
2925	0	36.4954430960341	0.49972088793654	0.50027911206346	0	OK	True Neg.
10694	0	36.4795779571031	0.499702456393005	0.500297543606995	0	OK	True Neg.
9065	0	36.4745518016055	0.499696617187837	0.500303382812163	0	OK	True Neg.

○ Logistic
○ ROC
○ Table
○ Distribution

+ Current Model
− General

Total Cases	18253
Quantiles	10
Positive Cases	11721 (64.21%)
Negative Cases	6532 (35.79%)

+ Quantiles
+ ROC
+ Cutoff Points
+ Gains
+ Lift
+ Log Odds
+ Logistic Fit
− Logistic Confusion Matrix

Logistic Cutoff Point	36.7356915940758
Correct	13549 (74.23%)
Wrong	4704 (25.77%)
Total	18253
TP	9338 (51.16%)
FN	2383 (13.06%)
FP	2321 (12.72%)

◀ | ▶

Logistic Regression

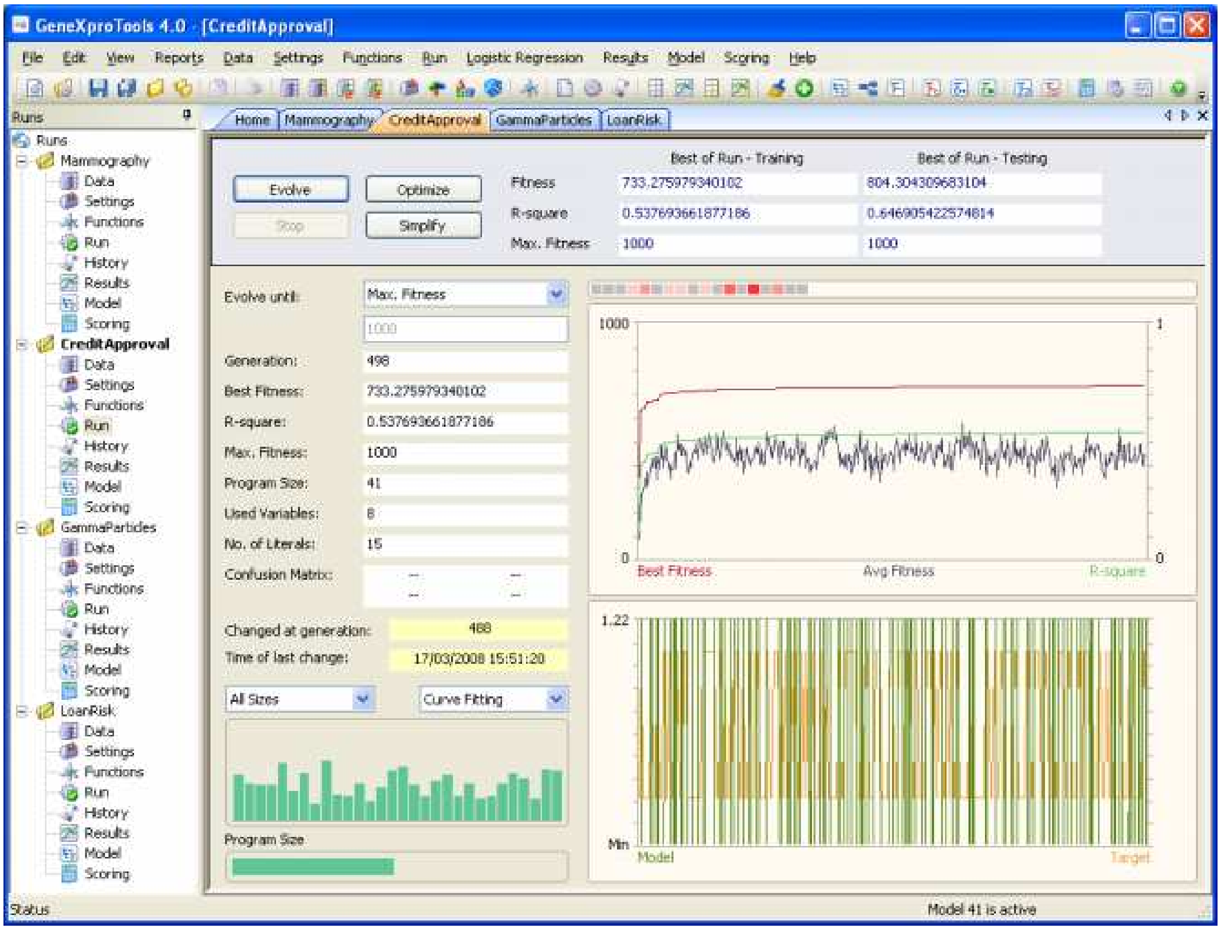
Class Encodings and Evolutionary Strategies

The addition of the **Logistic Regression Analytics Platform** to GeneXproTools 4.0 is a **client driven** release in response to specific user requests and the analysis of how GeneXproTools is being used in the wild. This first release of the Logistic Regression Analytics Platform leverages the strengths of the Function Finding infrastructure, adding significant analytics capabilities to GeneXproTools. Future versions of GeneXproTools will build on these new analytics tools improving further the user experience and expanding the scope of GeneXproTools.

To help you reap the benefits of the combination of GeneXproTools learning algorithms and the new statistical methods and analyses we suggest that you explore the following in your runs:

1. Create your core models in the Function Finding Framework
2. Use the R-square or the Correlation Coefficient fitness functions to drive evolution
3. Explore different Class Encodings to search the solution space

This implementation of the Logistic Regression Analytics Platform uses the **Function Finding Framework** in the model creation phase which was tweaked in some places. As our tests indicate that the R-square and the Correlation Coefficient are the most appropriate fitness functions for this type of analysis, when you create a new Function Finding run using a binary dependent variable we infer that you are creating a run for the Logistic Regression Framework and reset the fitness function to the Correlation Coefficient instead of RRSE.



	Best of Run - Training	Best of Run - Testing
Fitness	733.275979340102	804.304309683104
R-square	0.537693661877186	0.646905422574814
Max. Fitness	1000	1000

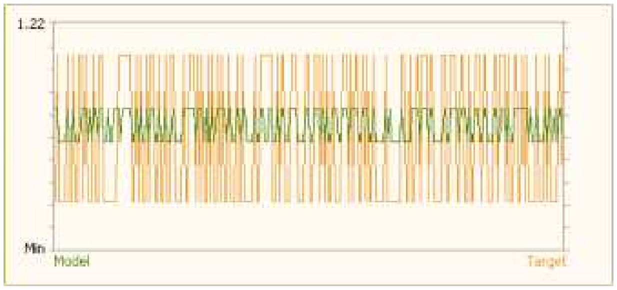
Evolve until: Max. Fitness
1000

Generation: 498
Best Fitness: 733.275979340102
R-square: 0.537693661877186
Max. Fitness: 1000
Program Size: 41
Used Variables: 8
No. of Literals: 15
Confusion Matrix: -- -- -- --

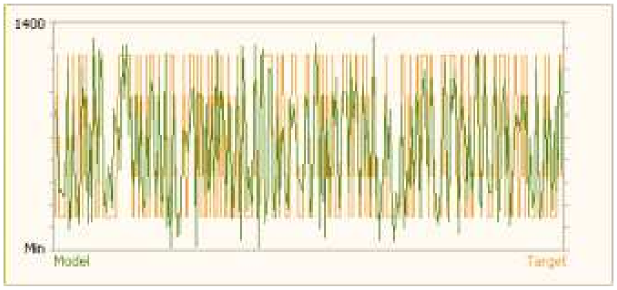
Changed at generation: 488
Time of last change: 17/03/2008 15:51:20

Status
Model 41 is active

The **Curve Fitting Chart** of the Run Panel, despite not being the most appropriate for a binary target, is still useful to get an idea of the kind of range the evolving models are exploring. Indeed, different fitness functions work on different ranges and therefore explore the solution space differently. Indeed, the reason why both the R-square and Correlation Coefficient fitness functions work so well with the standard 0/1 class encoding is that they can get free of the restricting 0/1 target range of the standard class encoding. For instance, a fitness function such as the one based on the Mean Squared Error (MSE) will only be able to drive evolution towards local optima around the boundaries of the standard 0/1 class encoding, like in the example below.



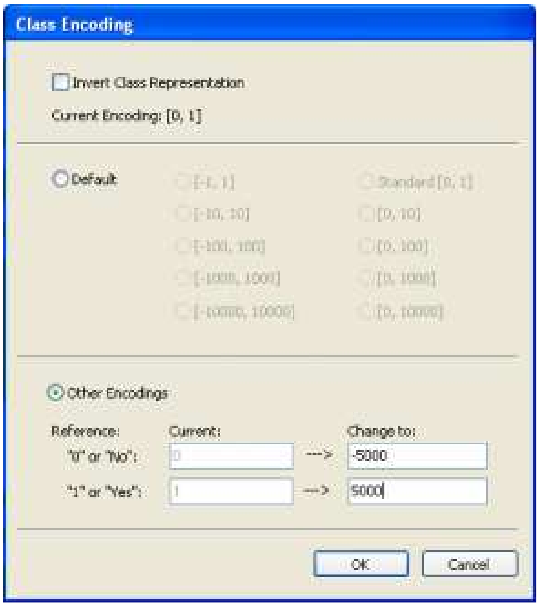
But if you use a different **Class Encoding**, say -1000/1000, you'll get a very different behavior, and although still confined to the target range, a fitness function such as the MSE has now much more room to explore and come up with good ranges for the model scores. This is of course the most important prerequisite for designing a good model. And you can observe this change in behavior straightaway in the Curve Fitting Chart.



For this new Release 2 of GeneXproTools 4.0, the implementation of the **R-square fitness function** was improved and strengthened and the **Correlation Coefficient fitness function** was redesigned using the absolute value of the Pearson Correlation Coefficient instead of the square. These two fitness functions appear to be the best creators of good models for the Logistic Regression Framework when using **standard 0/1 class encodings**. But as mentioned above, by using a different class encoding you'll see that other fitness functions besides the R-square and Correlation Coefficient start to work, giving excellent results too.

GeneXproTools allows you to **Change the Class Representation** easily and therefore you can experiment with different class encodings without much trouble (and you can just as easily revert to the standard 0/1 encoding if you feel more comfortable with it, although it has no bearing on the real meaning of the binary representation and how everything is processed and shown in the Logistic Regression Window, with the lowest value always representing the standard "0" or Negative cases, and the highest the standard "1" or Positive cases).

To change your Class Encoding within GeneXproTools, choose Change Class Encoding from the Logistic Regression menu. This opens the **Class Encoding Window**. In the Class Encoding Window, you can choose from several default values, but you can also experiment with all kinds of binary encodings, including systems with rational numbers, by entering any pair of two different numbers in the Change To box in the Other Encodings option.



Class Encoding

□ Invert Class Representation

Current Encoding: [0, 1]

○ Default

○ [-1, 1] ○ Standard [0, 1]
○ [-10, 10] ○ [0, 10]
○ [-100, 100] ○ [0, 100]
○ [-1000, 1000] ○ [0, 1000]
○ [-10000, 10000] ○ [0, 10000]

● Other Encodings

Reference:	Current:		Change to:
"0" or "No":	0	--->	-5000
"1" or "Yes":	1	--->	5000

Also notice that you can invert your class representation by ticking the **Invert Class Representation** check box. This means that what you had originally represented as “0” will become “1” and vice versa. This might prove useful in certain modeling situations, but please keep in mind that GeneXproTools will be handling what you originally had as negative cases as 1’s. And this means that within the Logistic Regression Framework all the predictions and analyses will be made for these new 1’s because the Logistic Regression Technique is by default designed to always predict the 1’s. Remember, however, that you can always revert to the original encoding by inverting the representation once more.

Also worth mentioning in this section about evolutionary strategies, is the fact that, in this Release 2 of GeneXproTools 4.0, we are for the first time allowing the **conversion** of Classification runs to the Function Finding Framework and vice versa (as long as the dependent variable in the Function Finding runs is binary, of course). This obviously means that you can explore all the fitness functions available for Classification (there are a total of 15 different ones, thus a nice complement to the basic R-square and Correlation Coefficient fitness functions of the Function Finding Framework) to evolve your models. Then, from the Logistic Regression Framework, you can access your model scores (they are not accessible in Classification mode as they are all rounded to either “0” or “1” according to pre-established 0/1 rounding thresholds). And once you have access to your model scores you are ready to proceed with the complete Logistic Regression Analytics, including the construction of Quantile Tables, analysis of Gains and Lift Charts, construction of ROC Curves and Cutoff Points Charts, and of course evaluation of the probabilities with Logistic Regression Models and also the construction of Logistic and ROC Confusion Matrixes.

Of course modelers interested in just **Discrete Classifications** can also benefit from this bridge between the Logistic Regression Framework and the Classification Framework. Perhaps the most important and obvious consequence of this interconnection, is the fact that, in the Logistic Regression Framework, the equivalent to the 0/1 Rounding Threshold of Classification models is being evolved totally **unsupervised** and therefore a whole new range of possibilities can be explored. Furthermore, in the Logistic Regression Framework, GeneXproTools also shows and computes the Confusion Matrix derived from the ROC Curve and Optimal Cutoff Point, and therefore you can see immediately how good a model created within the Logistic Regression Framework is at making crisp classifications. And you can make good use of that finding immediately by converting that run to Classification.

When a Logistic Regression run is converted to Classification, the Optimal Model Threshold evaluated for the training dataset with the active model is automatically set up as the 0/1 Rounding Threshold to be used in the new Classification run. Note, however, that in Logistic Regression virtually every single model has its own finely adapted Optimal Model Threshold; but now, after the conversion, only the active model will be able to keep its own. This obviously means that you can only expect the ROC Confusion Matrix you obtained for the active model on the training dataset to match exactly the Confusion Matrix you get on the Classification side. For the testing dataset, however, results may be slightly different. This shouldn’t be a problem, though, for a modeler is usually interested in the best-of-run model and also because it is within the expected variation one gets when a model is checked against unseen data. Note also that if you decide to use this imported model as seed to create even better models for discrete classifications, the whole population will now all adapt to the now shared and fixed 0/1 Rounding Threshold.

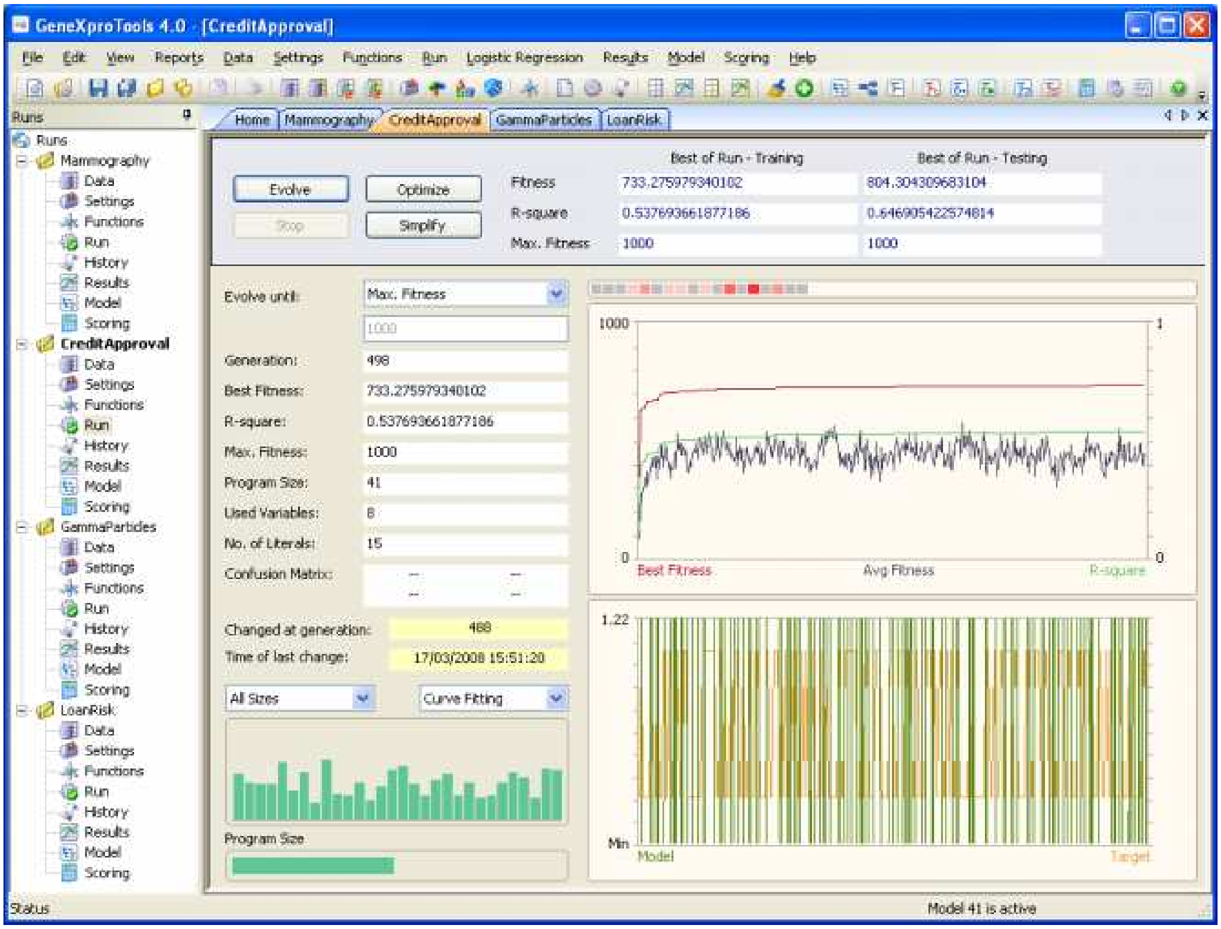
It is also worth pointing out that, when you convert a Logistic Regression run to Classification, you can also use the Logistic Cutoff Point as your 0/1 Rounding Threshold. Note, however, that in this case you'll have to set up the 0/1 Rounding Threshold manually in the Fitness Function Tab of the Settings Panel. The confusion matrix you'll get in this case on the Classification side will match obviously the Logistic Confusion Matrix.

◀ | ▶

Logistic Regression

Testing a Model

The **predictive accuracy** of logistic regression models (or more precisely, the core model of the logistic regression procedure) can be evaluated like all the models are evaluated in GeneXproTools, that is, as soon as evolution stops and if a testing set is available, both the fitness and R-square are immediately evaluated for the **testing dataset** and the results are shown straightaway on the **Run Panel**. Furthermore, an additional set of statistics, including the correlation coefficient, are evaluated and shown in the **Results Panel**. And there you can also test the predictive accuracy of all the models created in a run.



	Best of Run - Training	Best of Run - Testing
Fitness	733.275979340102	804.304309683104
R-square	0.537693661877186	0.646905422574814
Max. Fitness	1000	1000

Evolve until:	Max. Fitness
	1000
Generation:	498
Best Fitness:	733.275979340102
R-square:	0.537693661877186
Max. Fitness:	1000
Program Size:	41
Used Variables:	8
No. of Literals:	15
Confusion Matrix:	-- -- -- --
Changed at generation:	488
Time of last change:	17/03/2008 15:51:20

Model 41 is active

When the **fitness** and **R-square** obtained for the testing set are about the same as the values obtained for the training set (and when the partition of the data is done correctly, they usually are, for GEP models rarely, if ever, overfit the data), this is a good indicator that your model is a good one and therefore can be used to make good predictions, which, in this case, means using it to create the final Logistic Regression Model in the Logistic Regression Window.

Additionally, within the Logistic Regression Framework, GeneXproTools allows you to run the whole set of analytics tools on the testing dataset, including construction and analysis of Quantile Tables, ROC Curves, Cutoff Points, Gains and Lift Charts, Logistic Regression and Logistic Fit, and ROC and Logistic Confusion Matrixes. For that you just have to select Testing Set in the Dataset combo box in the Logistic Regression Window.

Note, however, that this **additional testing procedure** builds its own Quantile Table and also evaluates and uses its own slope and intercept for the Logistic Regression Model. This means obviously that the logistic regression parameters evaluated for the training dataset are not used in this testing procedure, which may have been interesting in some cases as a form of further testing the model.

If such a rigorous testing is desired, though, you can always perform a blind scoring on this testing dataset (you'll have to remove obviously the target output from the scoring dataset). Indeed, the logistic regression model that GeneXproTools deploys during scoring, uses the slope and intercept evaluated for the training dataset. This means that you can easily perform this precise testing within GeneXproTools using its new Logistic Regression Scoring Engine. The Scoring Engine was updated so that you could evaluate not only the predicted probabilities but also the most likely class. This means that you can use it to quickly compute rigorously the predictive accuracy by comparing the scoring results with your blind target values.

◀ | ▶

Logistic Regression

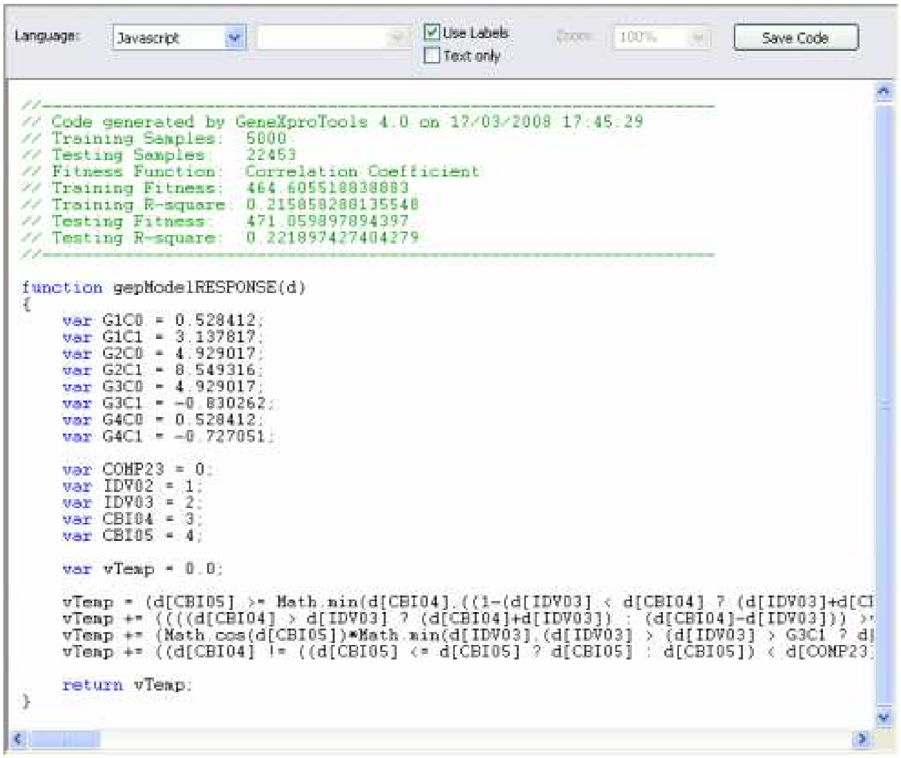
Making Categorical and Probabilistic Predictions

The goal in Logistic Regression is to assign probabilities to model scores, creating a reliable **ranking system** that can be used straightaway to evaluate the risk involved in financial and insurance applications, to rank potential respondents in a marketing campaign, or to evaluate the risk of contracting a disease.

The Logistic Regression Framework of GeneXproTools builds on the **model scores** it generates with its **Evolutionary Algorithms** and then applies the canonical **Logistic Regression Technique** to **estimate probabilities** for each model score. And once you know the probability of an event, you can also make **categorical predictions** and consequently infer easily crisp confusion matrixes.

Thus, with its new Logistic Regression Framework, GeneXproTools offers a highly robust hybrid system in which sophisticated multivariate nonlinear models are easily created by evolution and then empowered by traditional statistical modeling techniques.

GeneXproTools **scores new cases** using the **Javascript code** it generates for your model, but also adds (internally) the logistic regression equation with its slope and intercept to the code in order to evaluate the probabilities.


Language: Javascript

☑ Use Labels

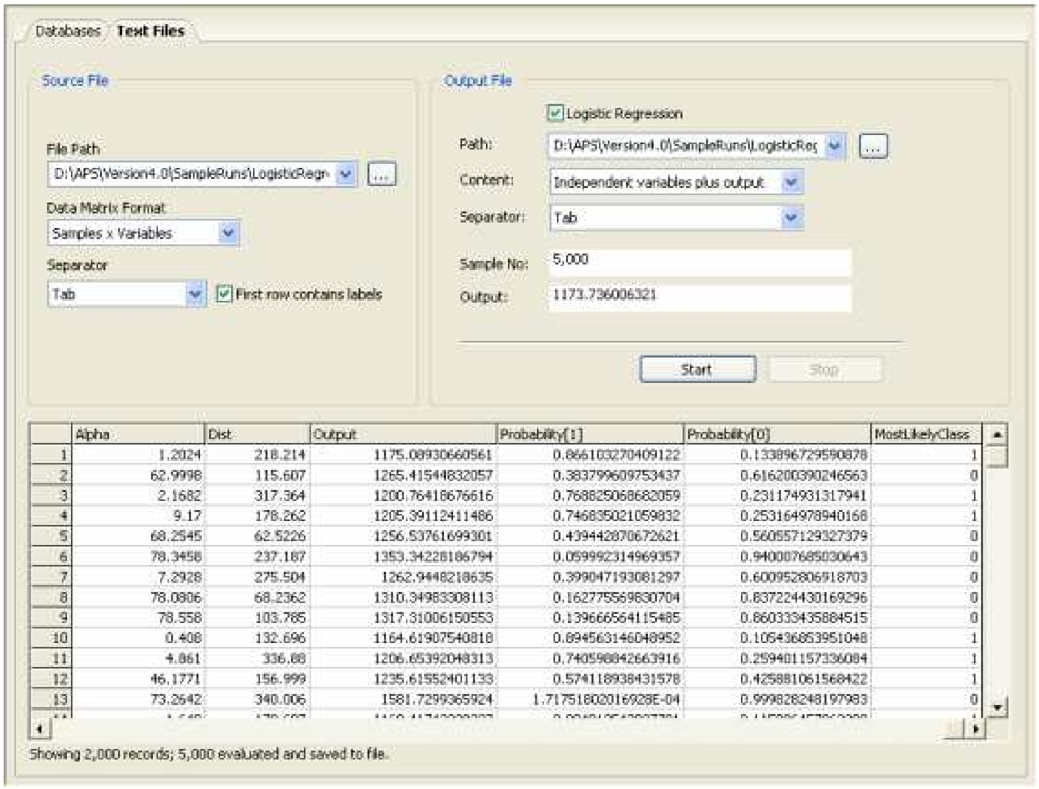
□ Text only

Zoom: 100%

```
//------------------------------------------------------------------------
// Code generated by GeneXproTools 4.0 on 17/03/2008 17:45:29
// Training Samples:   5000
// Testing Samples:    22453
// Fitness Function:   Correlation Coefficient
// Training Fitness:   464.605518838883
// Training R-square:  0.215858288135548
// Testing Fitness:    471.059897894397
// Testing R-square:   0.221897427404279
//------------------------------------------------------------------------

function gepModelRESPONSE(d)
{
    var G1C0 = 0.528412;
    var G1C1 = 3.137817;
    var G2C0 = 4.929017;
    var G2C1 = 8.549316;
    var G3C0 = 4.929017;
    var G3C1 = -0.830262;
    var G4C0 = 0.528412;
    var G4C1 = -0.727051;

    var COMP23 = 0;
    var IDV02 = 1;
    var IDV03 = 2;
    var CBI04 = 3;
    var CBI05 = 4;

    var vTemp = 0.0;

    vTemp = (d[CBI05] >= Math.min(d[CBI04],((1-(d[IDV03] < d[CBI04] ? (d[IDV03]+d[C
    vTemp += ((((d[CBI04] > d[IDV03] ? (d[CBI04]+d[IDV03]) : (d[CBI04]-d[IDV03])) >
    vTemp += (Math.cos(d[CBI05])*Math.min(d[IDV03],(d[IDV03] > (d[IDV03] > G3C1 ? d
    vTemp += ((d[CBI04] != ((d[CBI05] <= d[CBI05] ? d[CBI05] : d[CBI05]) < d[COMP23

    return vTemp;
}
```

In order to score new cases with the Scoring Engine of GeneXproTools you need to:

1. Go to the Scoring Panel and tick the Logistic Regression check box.
   In case you haven't yet evaluated the logistic regression parameters (the slope and intercept of the logistic regression model) in the Logistic Regression Window, GeneXproTools prompts you to go there so that it can perform these calculations.
2. Enter the path for the scoring data or connect to the database where your new cases are kept.
3. Enter the path for the file in which the scoring results will be saved.
   If you also want to include the input values in the output file, you have to choose Independent Variables Plus Output in the Content combo box.
4. Press the Start button to score your new cases.
   GeneXproTools shows the scoring results for the first 2000 cases in the Scoring Table of the Scoring Panel for a quick preview. All the scoring results, however, are saved to file. They include, besides the values of the input variables, the model output, probability[1] and probability[0], and the most likely class.



File Path: D:\APS\Version4.0\SampleRuns\LogisticRegr

Data Matrix Format: Samples x Variables

Separator: Tab

☑ First row contains labels

☑ Logistic Regression

Path: D:\APS\Version4.0\SampleRuns\LogisticReg

Content: Independent variables plus output

Separator: Tab

Sample No: 5,000

Output: 1173.736006321

	Alpha	Dist	Output	Probability[1]	Probability[0]	MostLikelyClass
1	1.2024	218.214	1175.08930660561	0.866103270409122	0.133896729590878	1
2	62.9998	115.607	1265.41544832057	0.383799609753437	0.616200390246563	0
3	2.1682	317.364	1200.76418676616	0.768825068682059	0.231174931317941	1
4	9.17	178.262	1205.39112411486	0.746835021059832	0.253164978940168	1
5	68.2545	62.5226	1256.53761699301	0.439442870672621	0.560557129327379	0
6	78.3458	237.187	1353.34228186794	0.059992314969357	0.940007685030643	0
7	7.2928	275.504	1262.9448218635	0.399047193081297	0.600952806918703	0
8	78.0806	68.2362	1310.34983308113	0.162775569830704	0.837224430169296	0
9	78.558	103.785	1317.31006150553	0.139666564115485	0.860333435884515	0
10	0.408	132.696	1164.61907540818	0.894563146048952	0.105436853951048	1
11	4.861	336.88	1206.65392048313	0.740598842663916	0.259401157336084	1
12	46.1771	156.999	1235.61552401133	0.574118938431578	0.425881061568422	1
13	73.2642	340.006	1581.7299365924	1.71751802016928E-04	0.999828248197983	0

Showing 2,000 records; 5,000 evaluated and saved to file.

The Scoring Engine of GeneXproTools allows you to score as many new cases as you wish without leaving the GeneXproTools environment. But you can also score your new cases outside GeneXproTools using the code it automatically generates for your model in any of the **15 programming languages** it supports. Note, however, that you'll have to include the logistic regression equation with its slope and intercept so that you can compute the probabilities. For instance, to the **C++ code** GeneXproTools generates for your models, you could add the following code (highlighted in bold dark brown):

```
#include <math.h>

double gepModel(double d[], double* probabilityOne, double* probabilityZero, int*
mostLikelyClass);

double gepModel(double d[], double* probabilityOne, double* probabilityZero, int*
mostLikelyClass)
{
    *probabilityOne = 0.0;
    *probabilityZero = 0.0;
```

```
        *mostLikelyClass = 0;

        const double SLOPE = 2.10292613251647E-03;
        const double INTERCEPT = 0.169988753428363;

        const double G2C1 = -5.02304;

        double dblTemp = 0.0;

        dblTemp = (d[0]+pow(atan(pow(((d[3]+d[2])+d[4]),3)),3));
        dblTemp += ((d[0]+(sin((pow(G2C1,2)*(d[2]-d[4])))*d[2]))+d[2]);
        dblTemp += (d[0]+d[4]);

        *probabilityOne = 1.0 / (1.0 + exp(-(SLOPE * dblTemp + INTERCEPT)));
        *probabilityZero = 1.0 - (*probabilityOne);

        *mostLikelyClass = ((*probabilityOne) >= 0.5 ? 1 : 0);

        return dblTemp;
    }
```

Logistic Regression

Converting Classification Runs to Logistic Regression

GeneXproTools allows you to **convert** runs created within the Classification Framework to Logistic Regression. This means that you'll be able not only to access the raw scores of your classification models (that is, the model scores before they were rounded to either 0 or 1) but also to use them to assign probabilities to your categorical classifications in the Logistic Regression Framework.

Any old Classification run can be converted to Logistic Regression. But you may also consider creating new ones with the sole purpose of exploring all the Classification fitness functions (there are a total of 15 different fitness functions in the Classification Framework, therefore a nice addition to the basic R-square and Correlation Coefficient fitness functions of the Logistic Regression Platform). Then you can see how good these models perform in the Logistic Regression Framework by analyzing their ROC Curves and Optimal Cutoff Points, Quantile Tables and Gains and Lift Charts, and also their Logistic Fit and Confusion Matrixes. And if they are not yet what you need, you can always use them as seed (either here in the Function Finding Platform or back in the Classification Platform) to create better models with them. You can obviously repeat this process for as long as you wish, until you obtain the right model for your data.

To convert a Classification run to Logistic Regression you need to:

1. Within the Classification Framework, choose Convert To Logistic Regression in the Logistic Regression menu.
   This opens the Save As dialog box and also asks if you want to save the current run before converting it to Logistic Regression. This way you will be able to come back to it if you need to.
2. Type the run name for the new Logistic Regression run and then click Save.
   When you click Save, GeneXproTools takes you immediately to the Function Finding Framework. Note that some of the statistics of the models in the Run History are updated, showing now the R-square instead of the Classification Accuracy. Note, however, that these R-square values were evaluated using rounded model scores (0 or 1) in the Classification Framework and will, therefore, differ from the ones evaluated here using the raw model scores. By choosing Refresh All you can rapidly update these values to their true values in this new context. The default fitness function for the new run is the Correlation Coefficient as this fitness function, together with the similar R-square

fitness function, produces the best results with the standard 0/1 class encoding. But if you are using a different encoding, you may want to experiment with other fitness functions.

◀ | ▶

Logistic Regression

Converting Logistic Regression Runs to Classification

GeneXproTools also allows you to **convert Logistic Regression runs to Classification**. This means that, among other things, you can easily access all the Classification fitness functions to drive the evolutionary fitting (there are a total of 15 different fitness functions in the Classification Framework, which is a nice addition to the basic R-square and Correlation Coefficient fitness functions of the Logistic Regression Framework). By going back and forth between both platforms, you can explore different modeling tools to fine-tune your models.

But there is another advantage to being able to convert Logistic Regression runs to Classification, especially if your interests lay in crisp classifications alone.

In the Logistic Regression Framework, the 0/1 Rounding Threshold (which is called the Optimal Model Threshold in the Logistic Regression Framework) is entirely evolved and finely crafted by the evolutionary system, rather than being ad hoc imposed by the modeler. This capability of being able to evolve the Optimal Rounding Threshold might prove invaluable for modeling certain datasets, but, most importantly, we are dealing here with a much more flexible and adaptable system that is totally free to co-evolve a unique, finely adjusted, rounding threshold for each and every model.

When a Logistic Regression model is converted to Classification, the Optimal Model Threshold that is inferred from the ROC Curve and Optimal Cutoff Point Analysis, is automatically set up as the 0/1 Rounding Threshold of the new Classification run. Note, however, that this rounding threshold is finely adjusted to the training data that was used to create a particular model, more specifically, the active model of this run. This means that you can only expect the confusion matrix it now generates for the training dataset to match the ROC Confusion Matrix inferred in the Logistic Regression Framework. Note, however, that for inverted models, the confusion matrix you’ll get in the Classification Framework will be inverted relatively to the ROC Confusion Matrix, which was adjusted to match the configuration of the Logistic Confusion Matrix.

It is also worth pointing out that, when you convert a Logistic Regression run to Classification, you can also use the Logistic Cutoff Point as your 0/1 Rounding Threshold. Note, however, that in this case you'll have to set up the 0/1 Rounding Threshold manually in the Fitness Function Tab of the Settings Panel. The confusion matrix you'll get in this case on the Classification Framework will match obviously the Logistic Confusion Matrix.

To convert a Logistic Regression run to Classification you need to:

1. Within the Function Finding Framework, choose Convert To Classification in the Logistic Regression menu.
   This opens the Save As dialog box and also asks if you want to save the current run before converting it to Classification. This way you will be able to come back to it if you need to.
2. Type the run name for the new Classification run and then click Save.
   When you click Save, GeneXproTools takes you immediately to the Classification Framework. Note that, in the Run History, the fitness values that are shown there correspond to the ones evaluated in the Function Finding Framework. By choosing Refresh All you can rapidly update these values to their true values in this new context. The default fitness function for the new run is Squared Accuracy, but you can easily choose another one in the Fitness Function Tab of the Settings Panel.

Logistic Regression

Demo Functionality and Sample Runs

The **Demo** of GeneXproTools 4.0 allows you to experiment with **your own data** so that you can evaluate thoroughly the modeling capabilities of the software. Furthermore, in the Logistic Regression Framework, you have access to all the tools of the sophisticated **Analytics Platform** of this new Release, including:

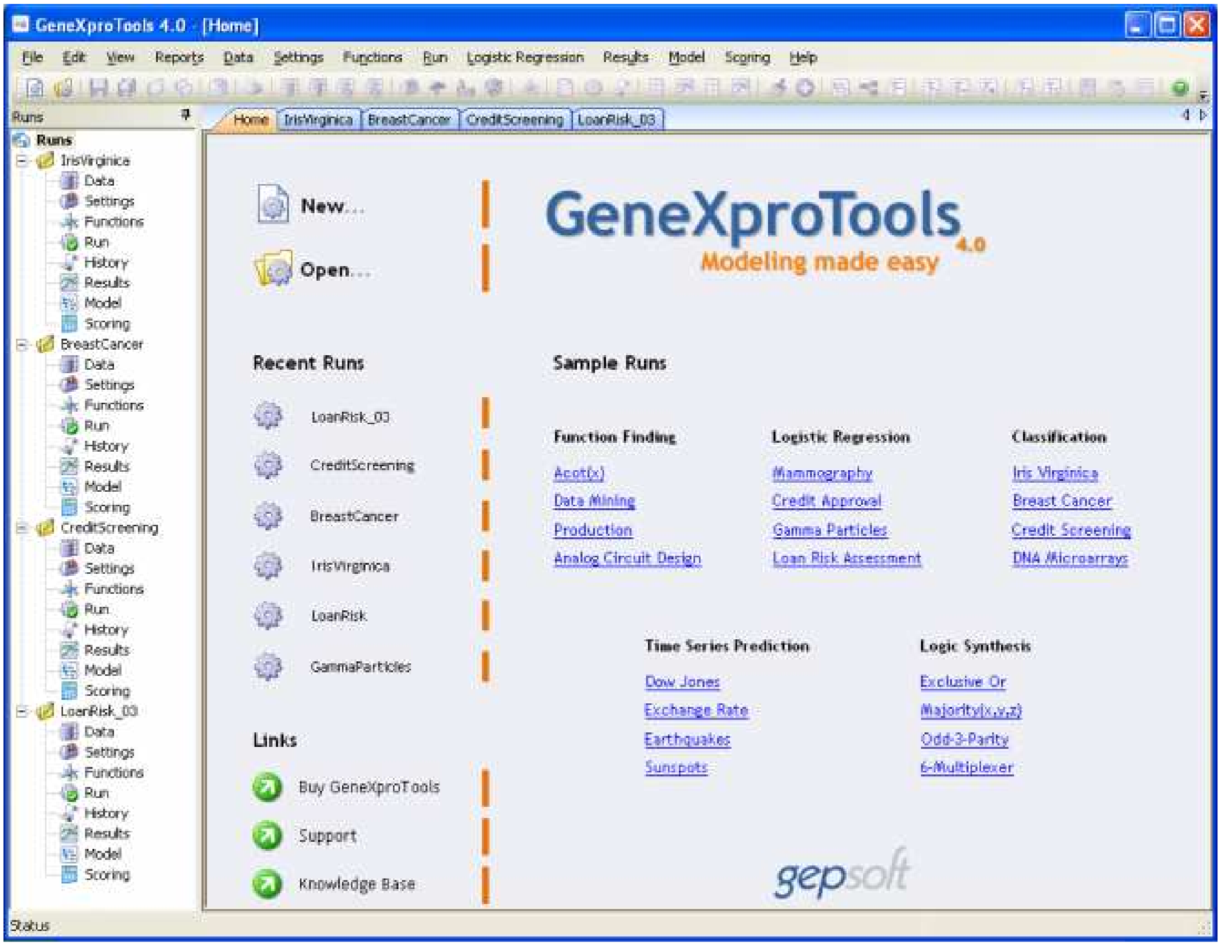
- Perform Quantile Analysis and Regression of your model scores
- Build and analyze ROC Curves
- Determine Optimal Cutoff Points for your model scores
- Perform detailed analyses of Gains and Lift Charts
- Create powerful Logistic Regression Models
- Assign probabilities to your model scores
- Build and analyze Logistic Fit Charts
- Infer and analyze Logistic and ROC Confusion Matrixes
- Test the predictive accuracy of your models on unseen data
- Explore vast solution spaces with a wide set of fitness functions
- Evolve models using different class encodings

Note, however, that with the Demo version you won't be able to see or access the **code** of the evolved models. And **scoring new cases** with your models is also ruled out as the built-in Scoring Engine of GeneXproTools is disabled in Demo mode. Furthermore, despite allowing the use of a testing set to evaluate the evolved models on unseen data, the Demo does not show the tables and charts in the Results Panel for the testing set. Notwithstanding, all the statistics used by GeneXproTools to evaluate the performance and predictive accuracy of the evolved models are available and fully functional both for the training and testing datasets.

In addition, in Demo mode you have access to a gallery of **Sample Runs** for Logistic Regression, for which all the features of GeneXproTools (code generation, visualization of parse trees, scoring, etc.) are totally functional:

- Identification of mammographic masses
- Credit card approval
- High energy gamma particles
- Loan risk assessment

To try any one of them, you just have to click in its link on the Welcome Screen of GeneXproTools.



Identification of Mammographic Masses

The goal here is to predict the severity (benign "0" or malignant "1") of a mammographic mass lesion from BI-RADS attributes (mass shape, mass margin, and mass density) and the patient's age.

The original real-world dataset (Elter et al. 2007) consists of 516 benign and 445 malignant masses that have been identified on full field digital mammograms collected at the Institute of Radiology of the University Erlangen-Nuremberg, between 2003 and 2006.

The training and testing datasets used in this run were generated by standard random partitioning of the original dataset. All missing values in the original dataset were replaced by the average. For training, a total of 640 cases were used, whereas the remaining 321 cases were used for testing.

The original dataset was obtained through the UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science. The file name is mammographic_masses.data and is available for download at:

http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass

Credit Card Approval

The goal in this problem is to decide whether to approve or not a customer’s request for a credit card. Each sample in the dataset represents a real credit card application and the output describes whether the bank granted the credit card or not.

All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. There are a total of 690 instances, 307 of which are successful "1" applications and 383 unsuccessful "0".

The training and testing datasets used in this run were generated by standard random partitioning of the original dataset. All missing values in the original dataset were replaced by the average. For training, a total of 460 cases were used, whereas the remaining 230 cases were used for testing.

A total of nine attributes in the original dataset are nominal and therefore had to be converted to numbers in order to be used in GeneXproTools.

Original encoding of nominal attributes:

A1: b, a.
A4: u, y, l, t (no t's were present in the donated dataset though).
A5: g, p, gg.
A6: c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff.
A7: v, h, bb, j, n, z, dd, ff, o.
A9: t, f.
A10: t, f.
A12: t, f.
A13: g, p, s.

Encoding used in this run:

A1: Binary, 1-column: b = 1, a = -1.
A4: Ternary, 1-column: u = -1, y = 1, l = 0.
A5: Ternary, 1-column: g = -1, p = 1, gg = 0.
A6: Ternary, 3-columns: c = (-1, -1, -1), d = (-1, -1, 0), cc = (-1, -1, 1), i = (-1, 0, -1), j = (-1, 0, 0), k = (-1, 0, 1), m = (-1, 1, -1), r = (-1, 1, 0), q = (-1, 1, 1), w = (0, -1, -1), x = (0, -1, 0), e = (0, -1, 1), aa = (0, 0, -1), ff = (0, 0, 0).
A7: Ternary, 2-columns: v = (-1, -1), h = (-1, 0), bb = (-1, 1), j = (0, -1), n = (0, 0), z = (0, 1), dd = (1, -1), ff = (1, 0), o = (1, 1).
A9: Binary, 1-column: t = 1, f = -1.
A10: Binary, 1-column: t = 1, f = -1.
A12: Binary, 1-column: t = 1, f = -1.
A13: Ternary, 1-column: g = -1, p = 0, s = 1.

The original dataset was obtained through the UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science. The file name is crx.data and is available for download at:

http://archive.ics.uci.edu/ml/datasets/Credit+Approval

High Energy Gamma Particles

In this problem, the goal is to discriminate shower images caused by primary gammas (signal or "1") from the images of hadronic showers initiated by cosmic rays in the upper atmosphere (background or "0").

The data are Monte Carlo generated to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. There are a total of 19020 instances in the original dataset, 12332 of which are gamma (signal) and 6688 hadron (background).

The training and testing datasets used in this run were prepared by standard random partitioning of the original dataset. For training, a total of 2000 instances were used, whereas the remaining 17020 instances were used for testing.

The original dataset was obtained through the UCI Machine Learning Repository, Irvine, CA: University of

California, School of Information and Computer Science. The file name is magic04.data and is available for download at:

http://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope

Loan Risk Assessment

The goal in this risk assessment problem is to decide whether to grant or not a customer’s request for a loan. Each sample in the dataset represents a real loan application and the output describes whether the customer paid "1" or defaulted "0".

All 27 attribute names and values in the original dataset have been changed to generic identifiers to protect confidentiality of the data. There are a total of 22453 loans in the original dataset, 14898 of which are good "1" and 7555 are bad "0".

The training and testing datasets used in this run were generated by standard random partitioning of the original dataset. All missing values in the original dataset were replaced by the average. For training, a total of 5000 cases were used, whereas the remaining 17453 cases were used for testing.

References

M. Elter, R. Schulz-Wendtland, and T. Wittenberg (2007). The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. *Medical Physics* 34(11), pp. 4164-4172.