Today, most organizations are stuck at lower-value descriptive analytics. But more sophisticated analysis can bring great business value.

Gartner says that three-quarters of companies plan to add more sophisticated predictive and prescriptive analytics in the future, which is known as advanced analytics.

However, data scientists have trouble applying advanced analytics. The main reasons for that are the high complexity of algorithms and their difficulty of use.

Neural Designer has been developed to meet that market needs: it implements advanced analytics algorithms and is very easy to use.

The software is based on neural networks, which are considered the most powerful technique for data analysis.

It contains a graphical user interface that clearly defines the workflow and provides comprehensive results.

In summary, Neural Designer allows you to get actionable insights resulting in smarter decisions and better business outcomes.
KEY BENEFITS

Easy to use

With Neural Designer there is no need for programming or building complicated block diagrams. Its user interface guides the user through a sequence of well-defined steps, in order to simplify data entry.

The software comes with many tutorials and examples to help you learn how to use it, and our team provides specialized technical support to our customers through telephone, email and Skype.

Great visualization

The Neural Viewer writes a report displaying comprehensive results from all tasks. Users can explore data and visualize the results with tables, charts and pictures that can be exported in order to be used in other tools.

Also, the whole report displayed in the viewer can be exported to PDF format. The objective is that the user is able to understand and interpret each step in the modeling process.
Advanced algorithms

Neural Designer contains a large range of advanced algorithms that allow data scientists to build the most powerful predictive models.

Sophisticated methods for data pre-processing, such as cleaning outliers or calculating principal components, are included.

The software implements neural network architectures with an arbitrary number of nonlinear layers, in order to build the most powerful predictive models.

It also includes different error and regularization methods so that the user can achieve the best results for a given data set.

Sophisticated training strategies, such as the quasi-Newton method or the Levenberg-Marquardt algorithm, have been developed for more accurate computations.

One of the main advantages of Neural Designer is the inclusion of an advanced model selection framework, which allows the user to obtain the most relevant variables in a given process.

The tool also contains many different methods for testing exhaustively the generalization capabilities of a predictive model, depending on the application at hand.

Finally, the resulting neural network can be operated within the program in different ways, or exported to any programming language, such as R or Python, using its mathematical expression.
High performance

Neural Designer outstands in terms of performance. It is developed in C++ for better memory management and higher processing speed, and implements CPU parallelization by means of OpenMP and GPU acceleration with CUDA.

The following plots show the results of a performance comparison among a predictive analytics software in R, a data mining platform in Java and Neural Designer.

As we can see, Neural Designer is able to analyze data sets up to 1000 times bigger, is up to 9 times faster and is up to 100 times more accurate its main competitors.

The data sets to make the comparisons are available at [www.artelnics.com/performancecomparison.zip](http://www.artelnics.com/performancecomparison.zip).

Simple deployment

Neural Designer provides an easy way for deploying predictive models. For that, you can use a standard such as PMML, or export the resulting model to programming languages such as R or Python.

An example of that expression in the R programming language is written below.

```r
expression <- function(x) {
  scaled_x<-2*(x+1)/(1+1)-1
  y_1_1<-tanh(0.361707-0.497807*scaled_x)
  y_1_2<-tanh(-0.15776-0.376231*scaled_x)
  y_1_3<-tanh(0.295148+0.493422*scaled_x)
  scaled_variable_2<-(-0.0531044-0.868102*y_1_1-0.778027*y_1_2+0.818376*y_1_3)
  outputs <- c(0.5*(scaled_variable_2+1.0)*(1+1)-1)
  outputs
}
```
## TECHNICAL FEATURES

### Application types
- Approximation (or modelling) to discover intricate relationships.
- Classification (or pattern recognition) to recognize complex patterns.
- Forecasting (or time series prediction) to predict current trends.
- Association (or auto-encoding) to learn a simplified representation of the data set.

### Data set
- Compatible with the most common data files: Excel, OpenOffice, CSV, Weka, DAT, TXT ...
- Also compatible with the most common data bases: Oracle, MySQL, SQLite, SQL Server and Access.
- Complete configuration of variables and instances.
- Exhaustive descriptive statistics.
- Estimation of variables importance by means of linear/logistic correlations.
- Advanced methods for data balancing.
- Innovative utilities for outlier detection.

### Neural network
- Network architecture with unlimited number of layers.
- Threshold, symmetric threshold, logistic, hyperbolic tangent and linear activation functions.
- Scaling and unscaling layers with minimum/maximum and mean/standard deviation methods.
- Probabilistic layer with binary and softmax methods.

### Loss index
- Sum squared error, mean squared error, root mean squared error and normalized squared error functionals for common data sets.
- Minkowski error for dealing with outliers.
- Cross-entropy error for pattern recognition.
- Weighted squared error for unbalanced data sets.
- Regularization for avoiding overfitting.

### Training strategy
- Gradient descent and conjugate gradient for training of big data sets.
- Quasi-Newton method for fast training of medium data sets.
- Levenberg-Marquardt algorithm for very fast training of small data sets.

### Model selection
- Incremental order and simulated annealing for finding the optimal network architecture.
- Growing inputs, pruning inputs and genetic algorithm for selecting the most important features.
| **Testing analysis** | • Complete error data and corresponding statistics calculation.  
| | • Linear regression analysis for function regression problems.  
| | • Confusion matrix for pattern recognition applications.  
| | • Full set of metrics for evaluation of binary classifiers.  
| | • ROC curve for diagnostic tests.  
| | • Cumulative gain and lift charts for segmentation applications in marketing.  
| | • Calibration plot for classification problems.  
| | • Error autocorrelation and cross-correlation for time series prediction.  
| | • List of misclassified instances.  |
| **Model deployment** | • Calculation of output values.  
| | • Directional plots for exploring the predictive model.  
| | • Jacobian values.  
| | • Exportable mathematical expression of the model.  
| | • Exportable predictive model in R and Python.  |
| **Output** | • Exhaustive results in an interactive report plenty of descriptions, tables and figures.  
| | • Report exportable to Word and Pdf formats.  
| | • Results exportable in data format.  |
| **Help** | • 6 extensive tutorials on the application of deep learning with Neural Designer.  
| | • 14 examples of machine learning applications in different fields.  
| | • Premium technical support by email, phone or video chat.  |
| **Performance** | • Software developed with the high performance programming language C++.  
| | • Code subjected to optimization techniques for memory management and processing speed.  
| | • CPU parallelization by means of OpenMP.  
| | • GPU acceleration with CUDA.  |
| **Supported platforms** | • Windows.  
| | • Mac OS X.  
| | • Linux (Debian and Ubuntu).  |
You can discover the power of Neural Designer by downloading a 15 day free trial from www.neuraldesigner.com.