**Business Intelligence Lab**

**Experiment 05**

**Aim:**

To explore Rapid Miner and implement classification model like Decision Tree and Naive Bayes etc

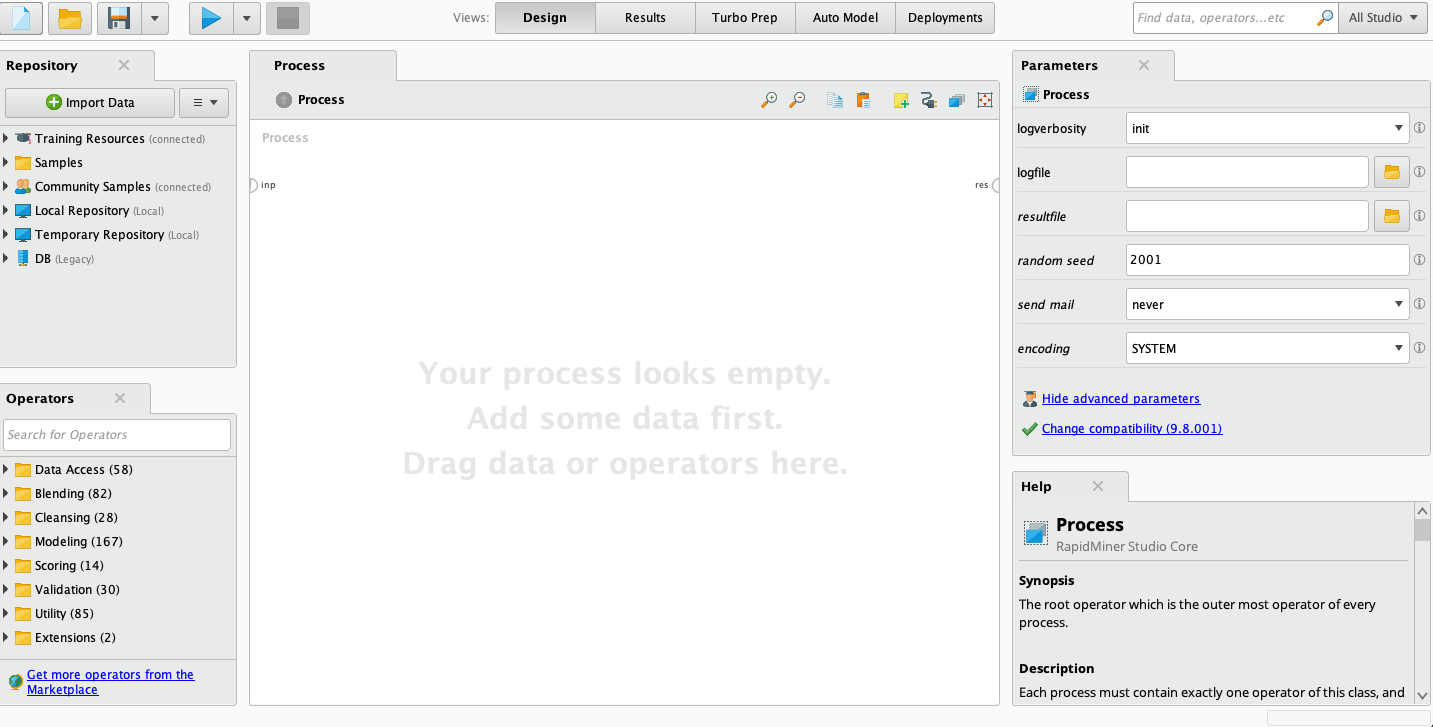
**Theory:**

**Rapidminer:**

Rapidminer is a comprehensive data science platform with visual workflow design and full automation. It means that we don’t have to do the coding for data mining tasks. Rapidminer is one of the most popular data science tools.

This is the graphical user interface of the blank process in rapidminer. It has the repository that holds our dataset. We can import our own datasets. It also offers many public datasets that we can try. We can also work with a database connection.

Below the repository window, it has an operator. The operators include everything we need to build a data mining process, such as data access, data cleansing, modeling, validation, and scoring. On the right is a parameters window. The parameters window is to adjust the operators.



Rapidminer can be downloaded from their official website (https://rapidminer.com/). It has a free version with limited functionality. The free version includes 10,000 data rows and 1 Logical Processor.

They also offer an educational program. So that students, professors, instructors, and researchers can have a free educational license for free.

**Classification**

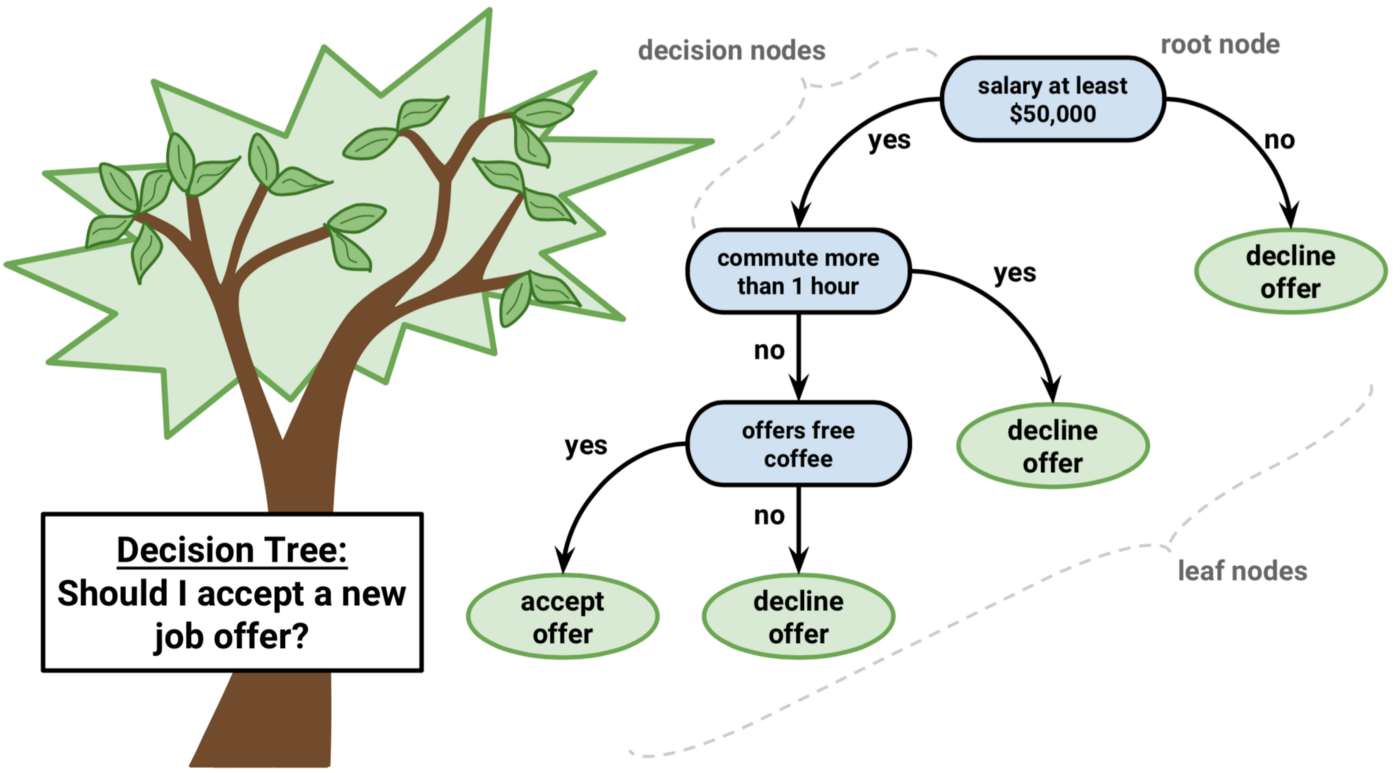
Classification is the process of predicting the class of given data points. Classes are sometimes called targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

**Classification algorithms**

There are a lot of classification algorithms available now but it is not possible to conclude which one is superior to the other. It depends on the application and nature of the available data set. For example, if the classes are linearly separable, the linear classifiers like Logistic regression, Fisher’s linear discriminant can outperform sophisticated models and vice versa.

**Decision Tree**

Decision trees build classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed. This process is continued on the training set until meeting a termination condition.

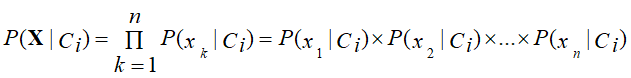


The tree is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact in the classification and they are identified using the information gain concept.

A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers. An over-fitted model has a very poor performance on the unseen data even though it gives an impressive performance on training data. This can be avoided by pre-pruning which halts tree construction early or post-pruning which removes branches from the fully grown tree.

**Naive Bayes**

Naive Bayes is a probabilistic classifier inspired by the Bayes theorem under a simple assumption that the attributes are conditionally independent.

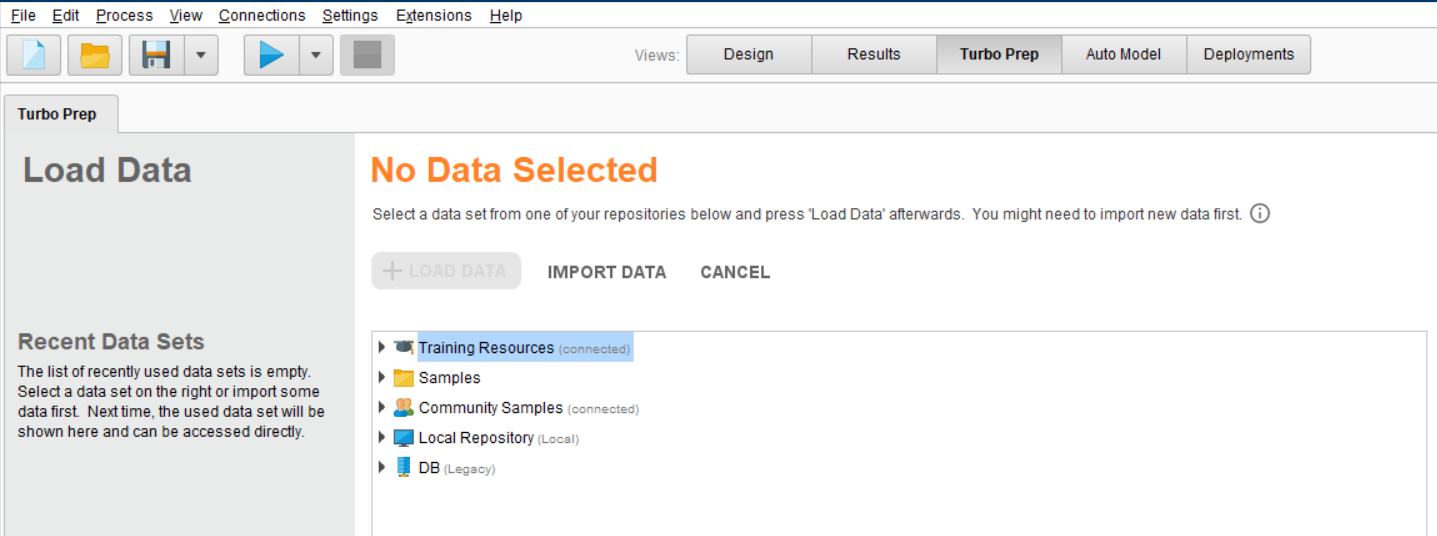


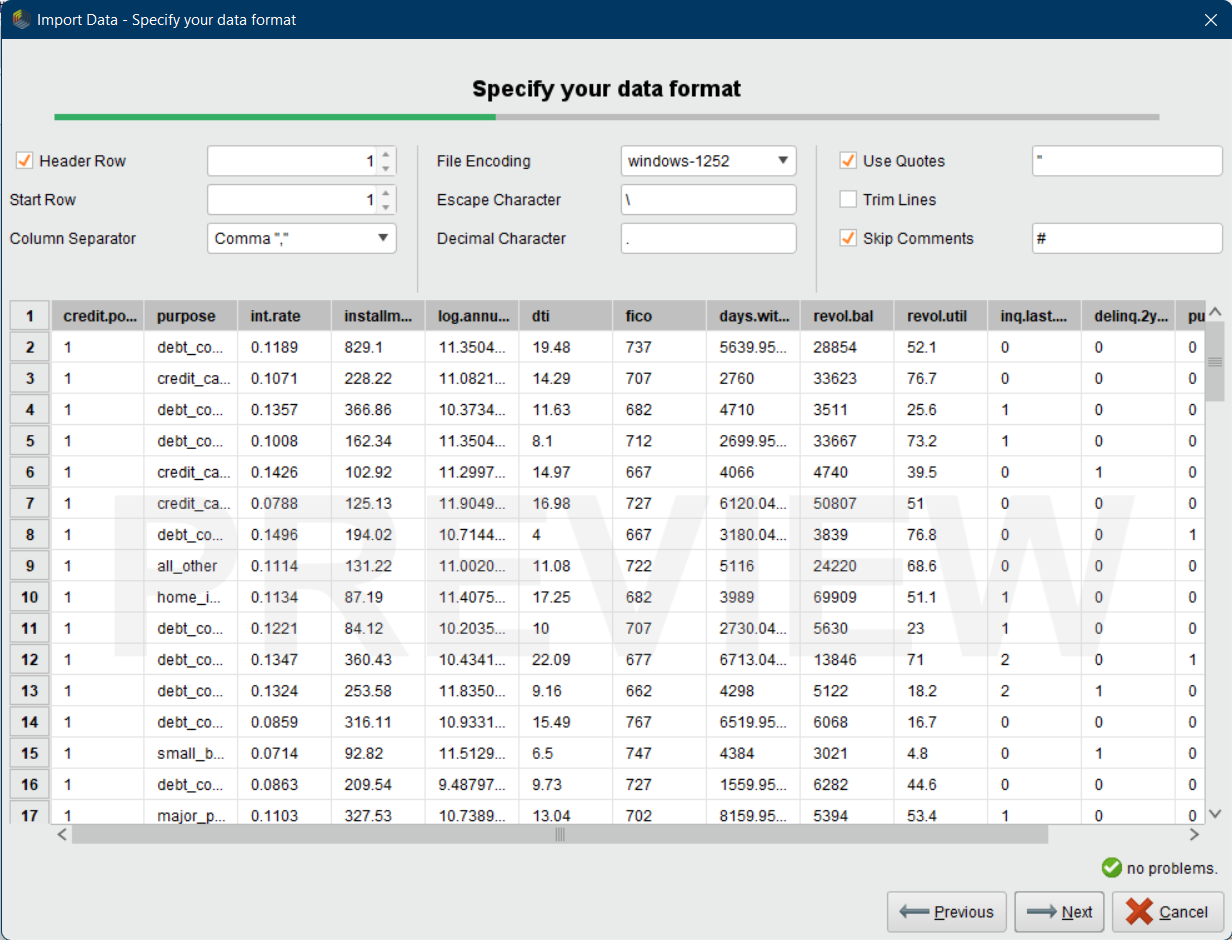
The classification is conducted by deriving the maximum posterior which is the maximal P(Ci|X) with the above assumption applying to Bayes theorem. This assumption greatly reduces the computational cost by only counting the class distribution. Even though the assumption is not valid in most cases since the attributes are dependent, surprisingly Naive Bayes was able to perform impressively.

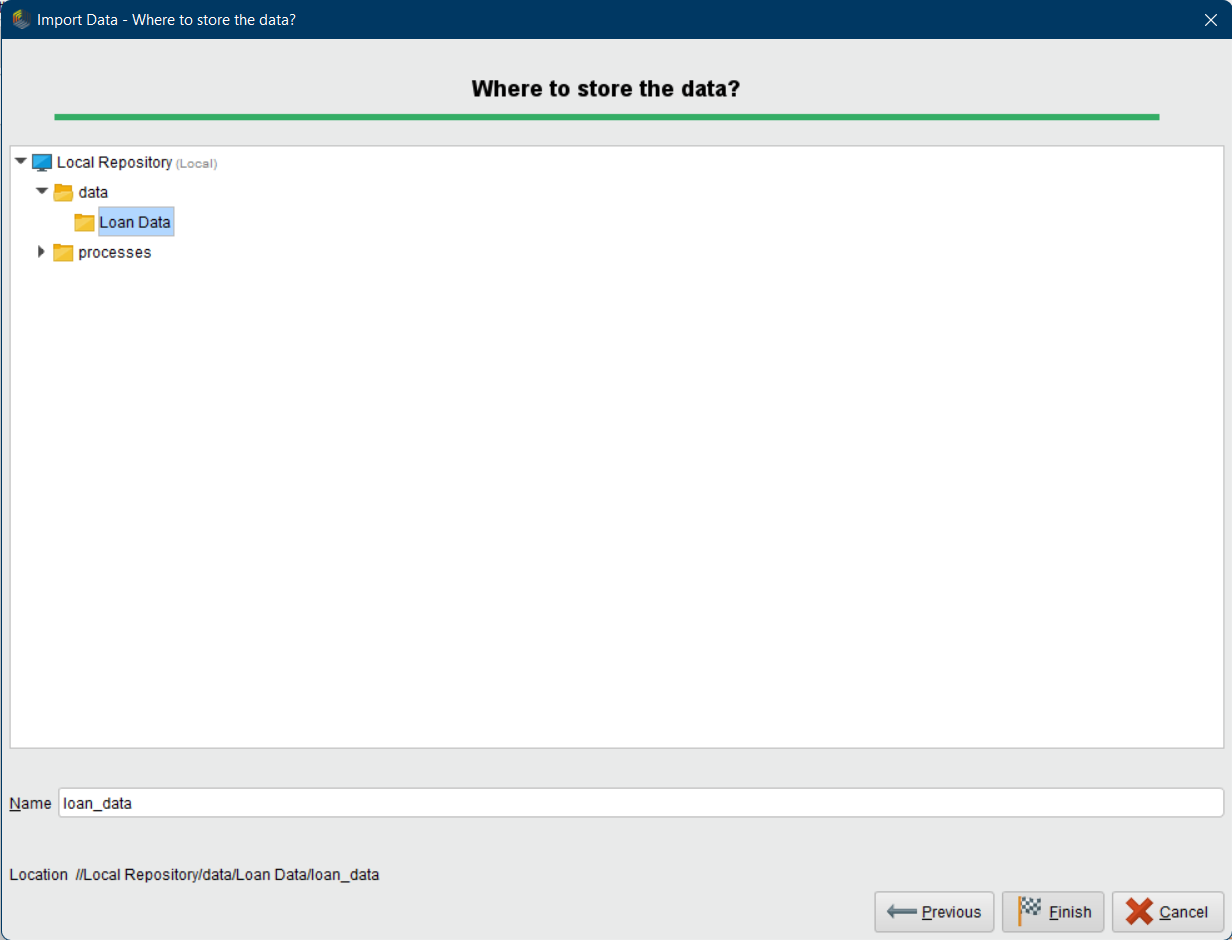
Naive Bayes is a very simple algorithm to implement and good results have been obtained in most cases. It can be easily scalable to larger datasets since it takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

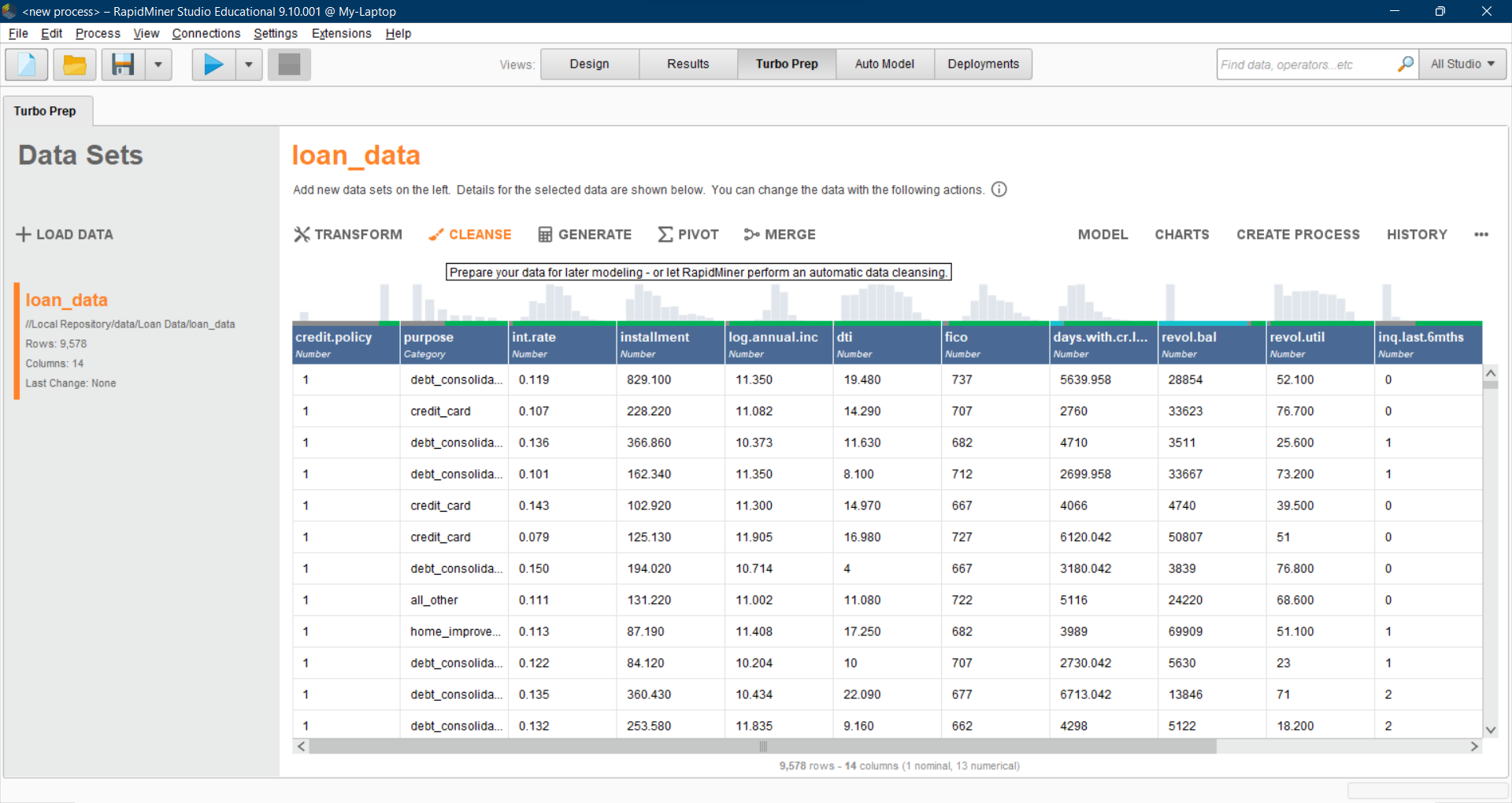
Naive Bayes can suffer from a problem called the zero probability problem. When the conditional probability is zero for a particular attribute, it fails to give a valid prediction. This needs to be fixed explicitly using a Laplacian estimator.

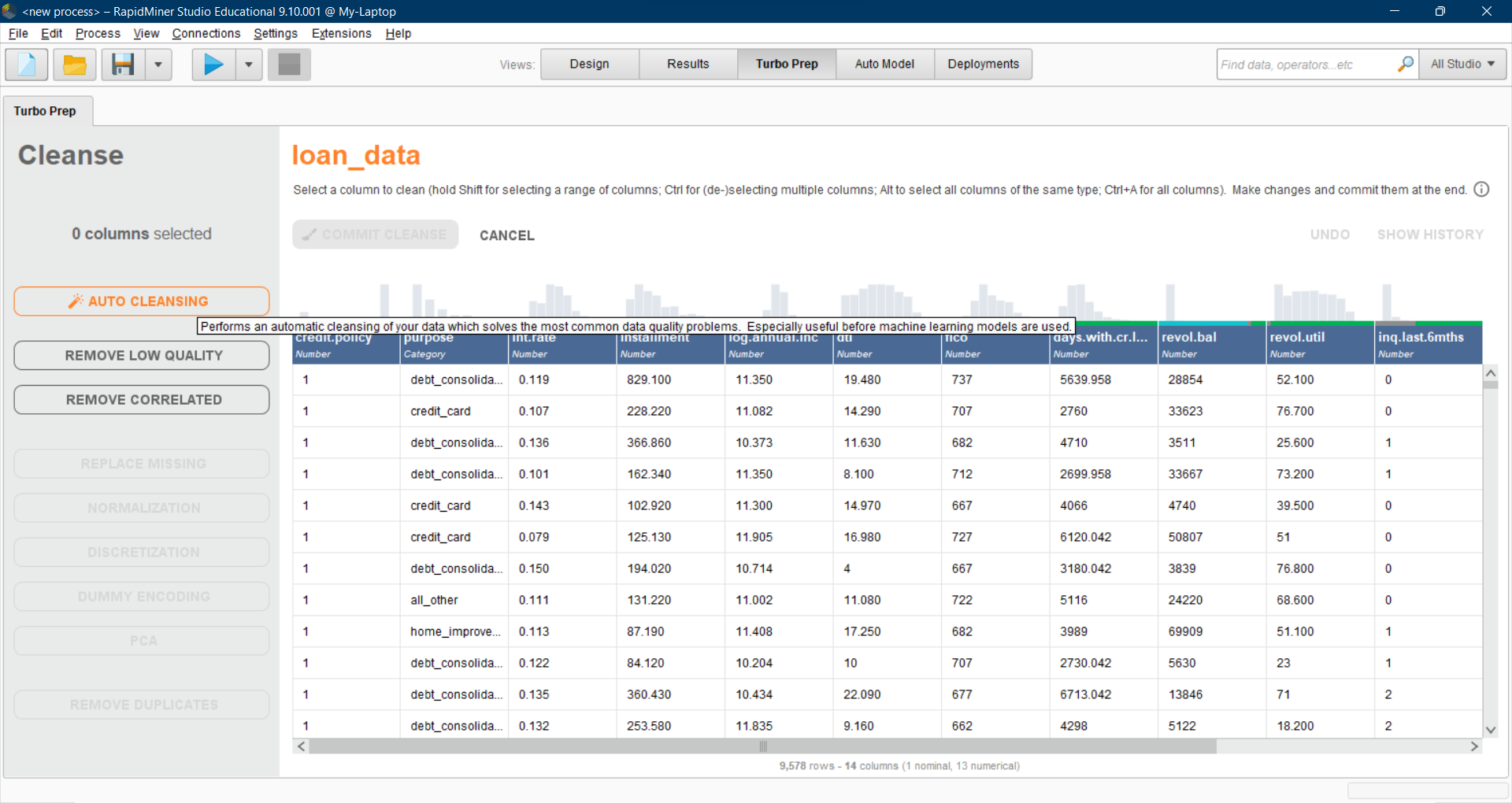
**Data Preprocessing in RapidMiner:**

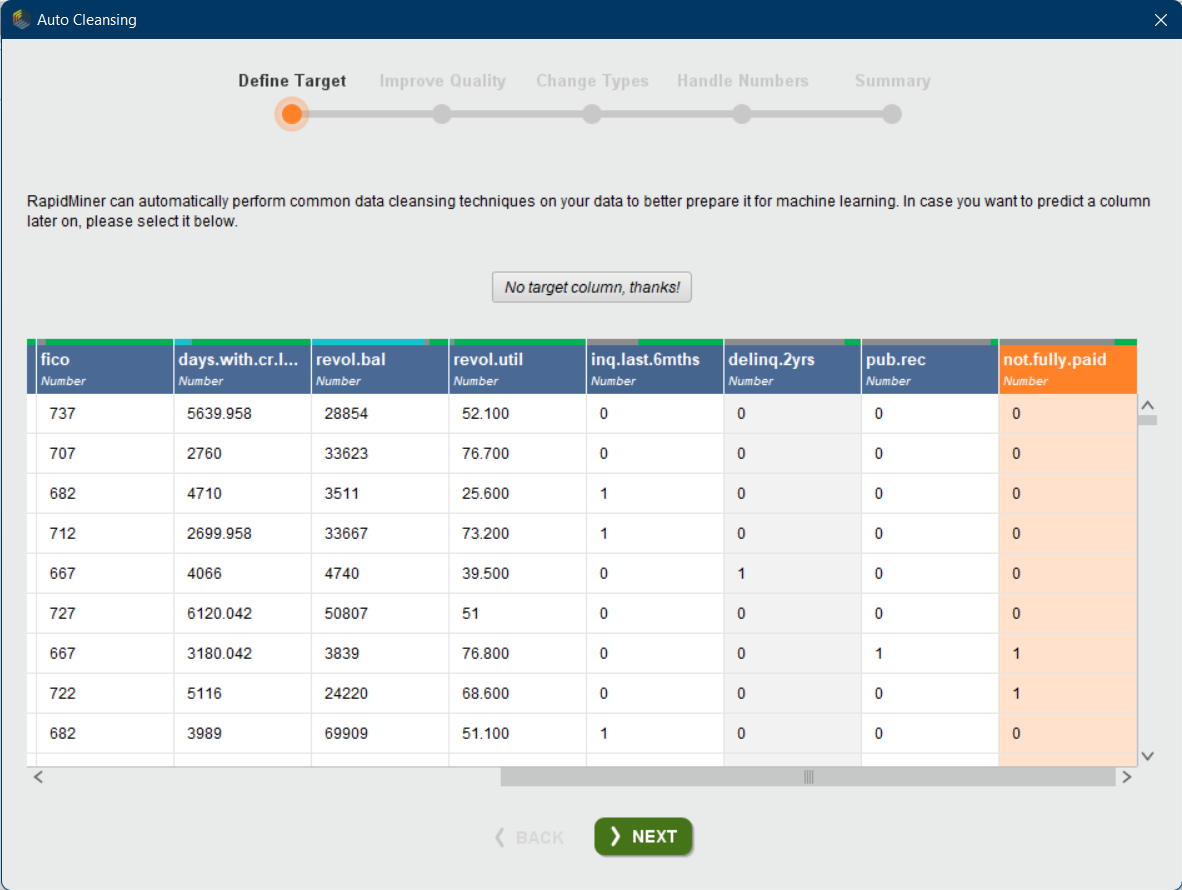


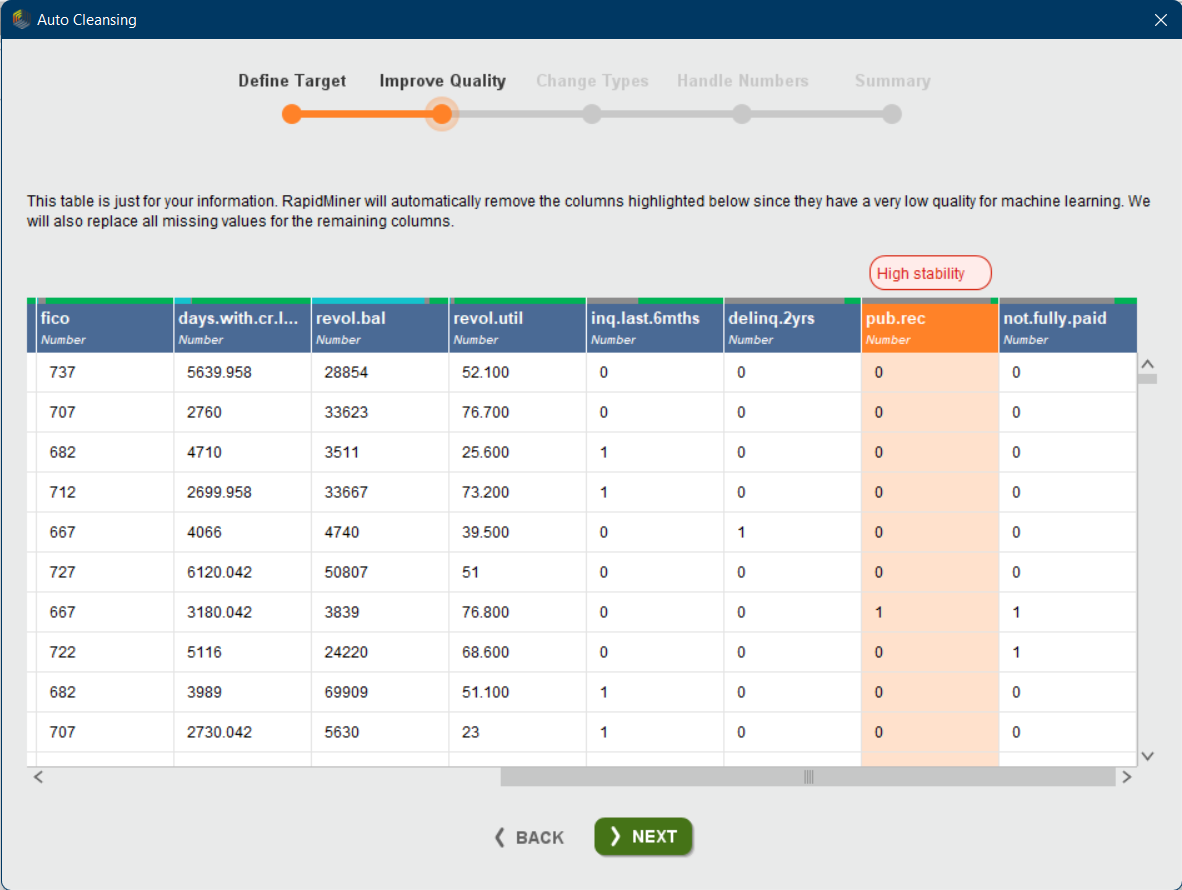


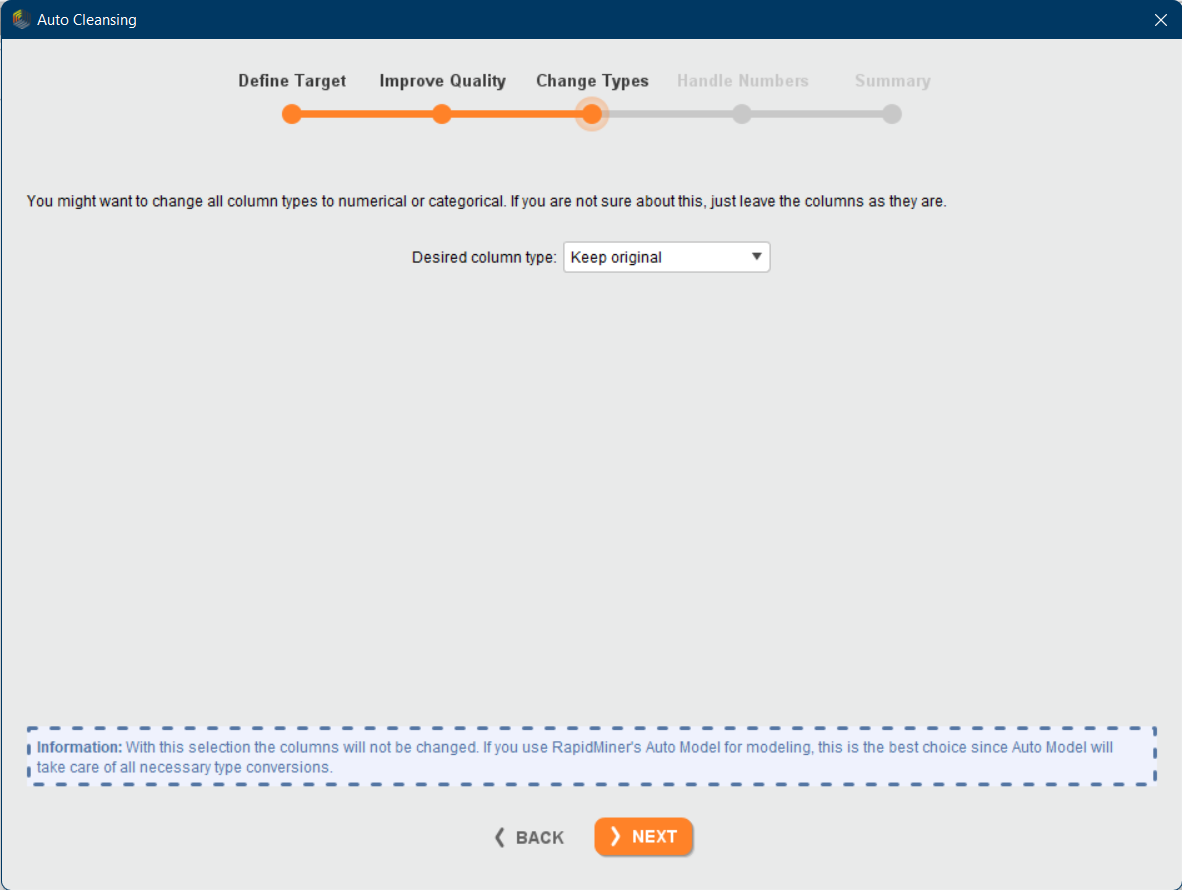


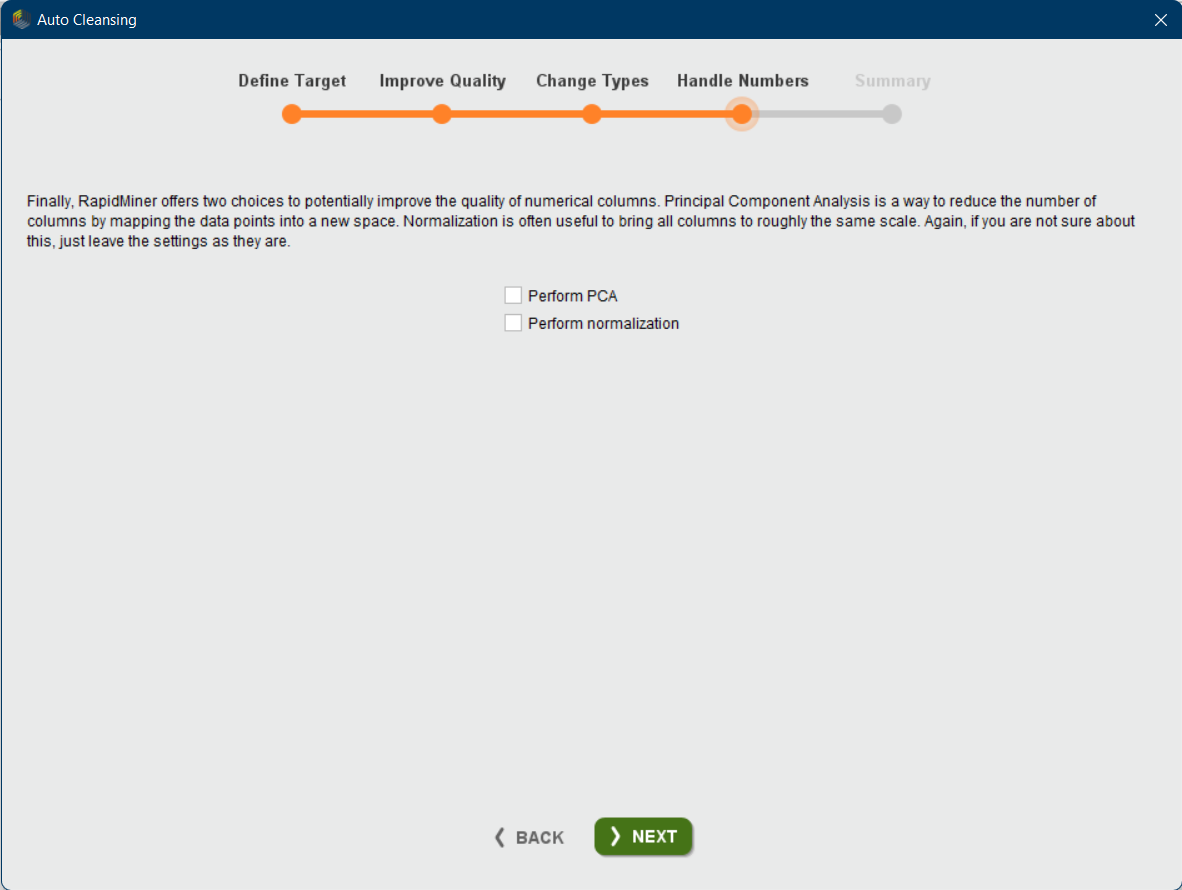


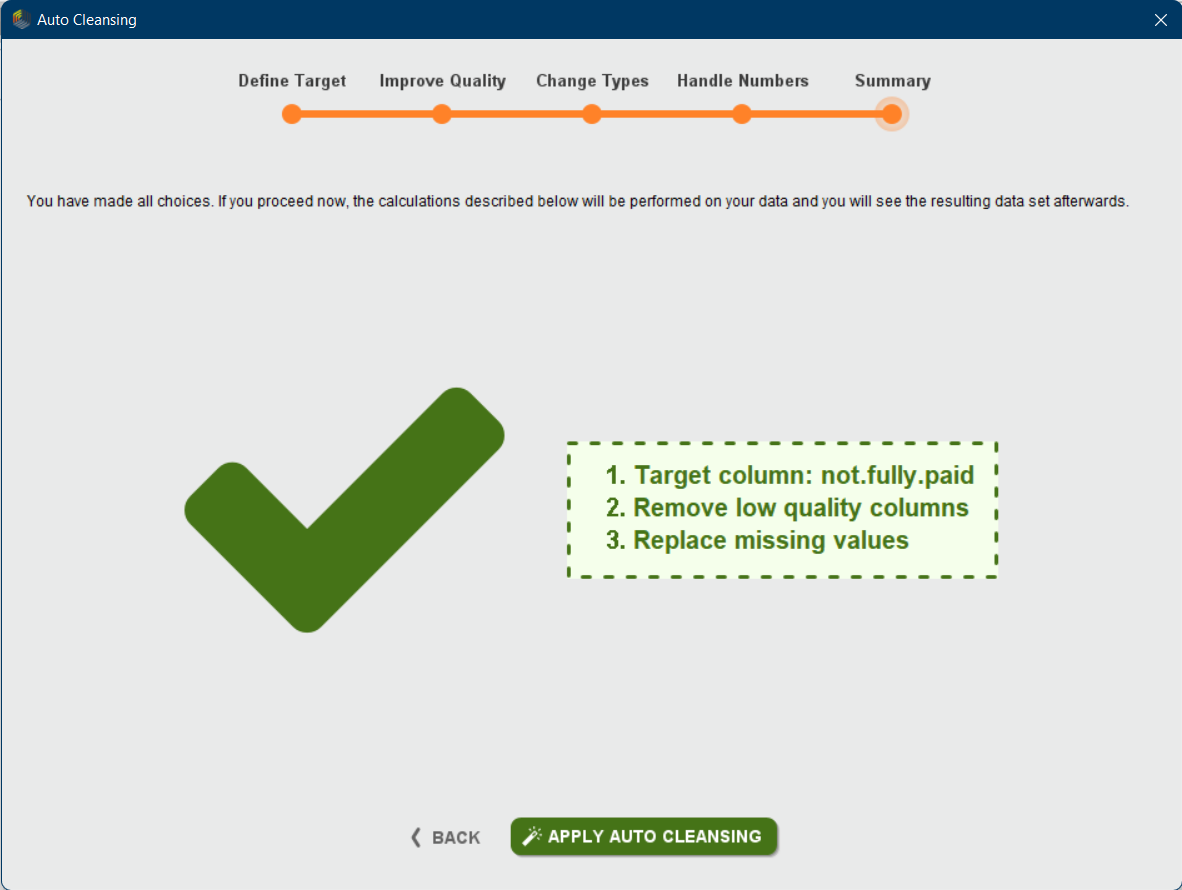




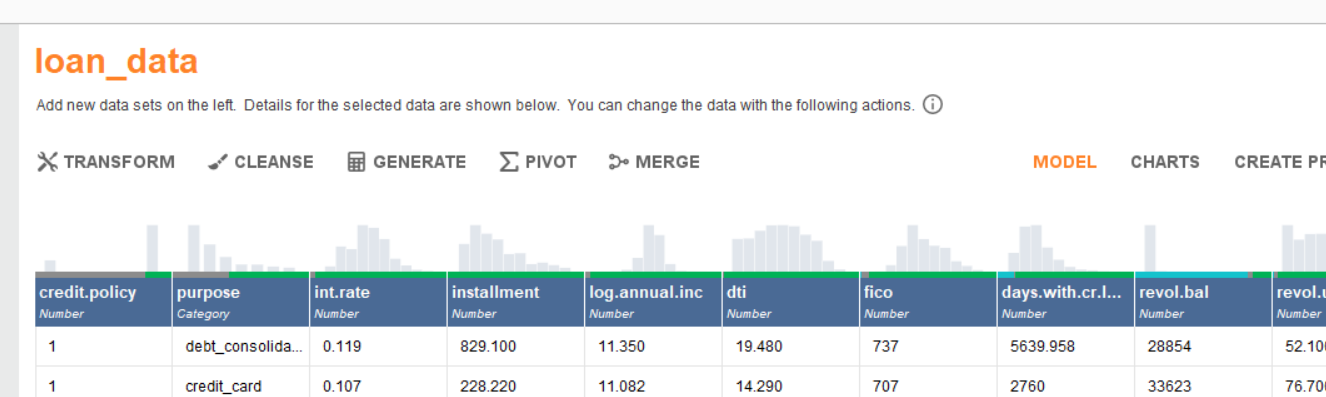


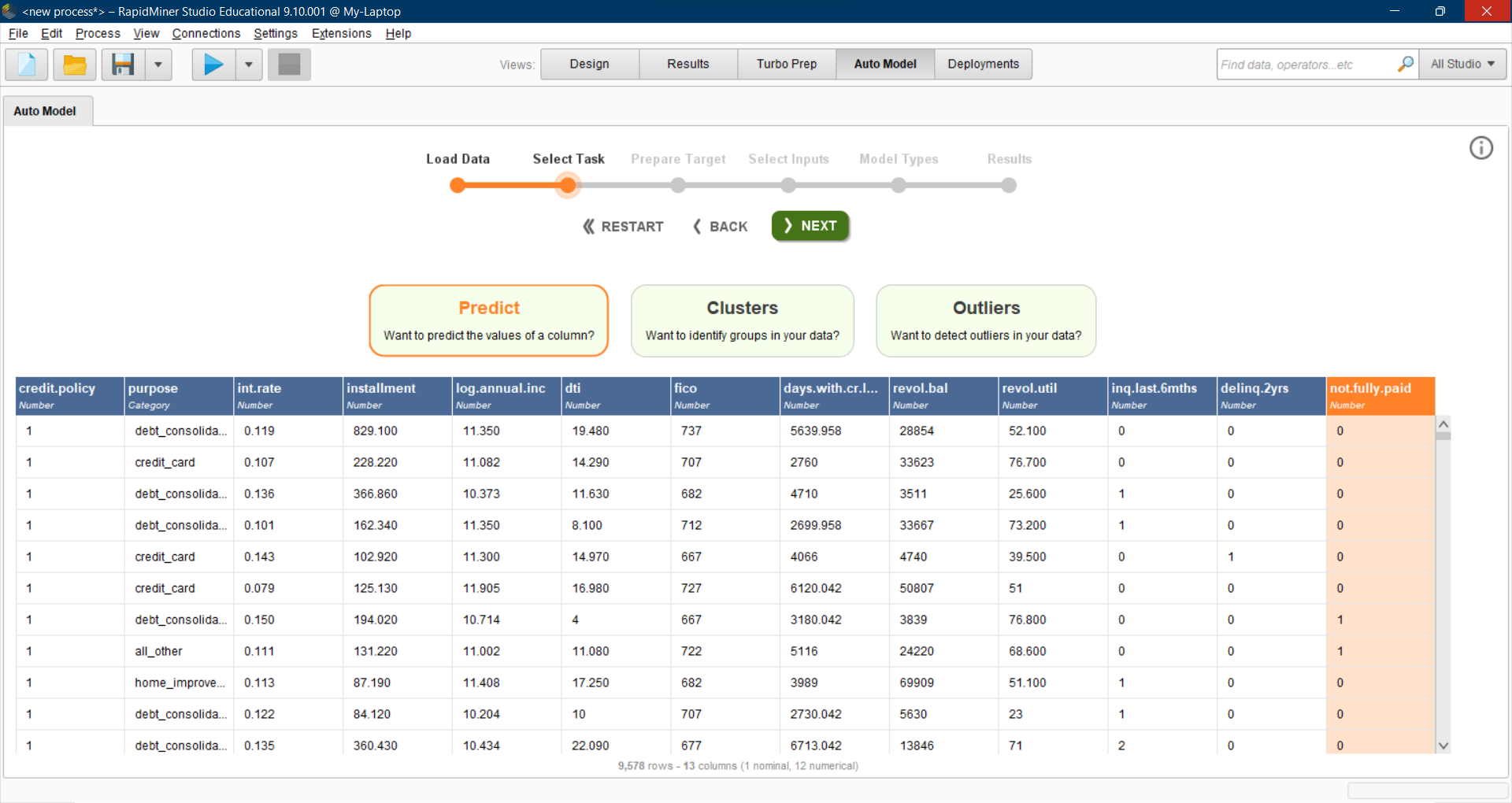


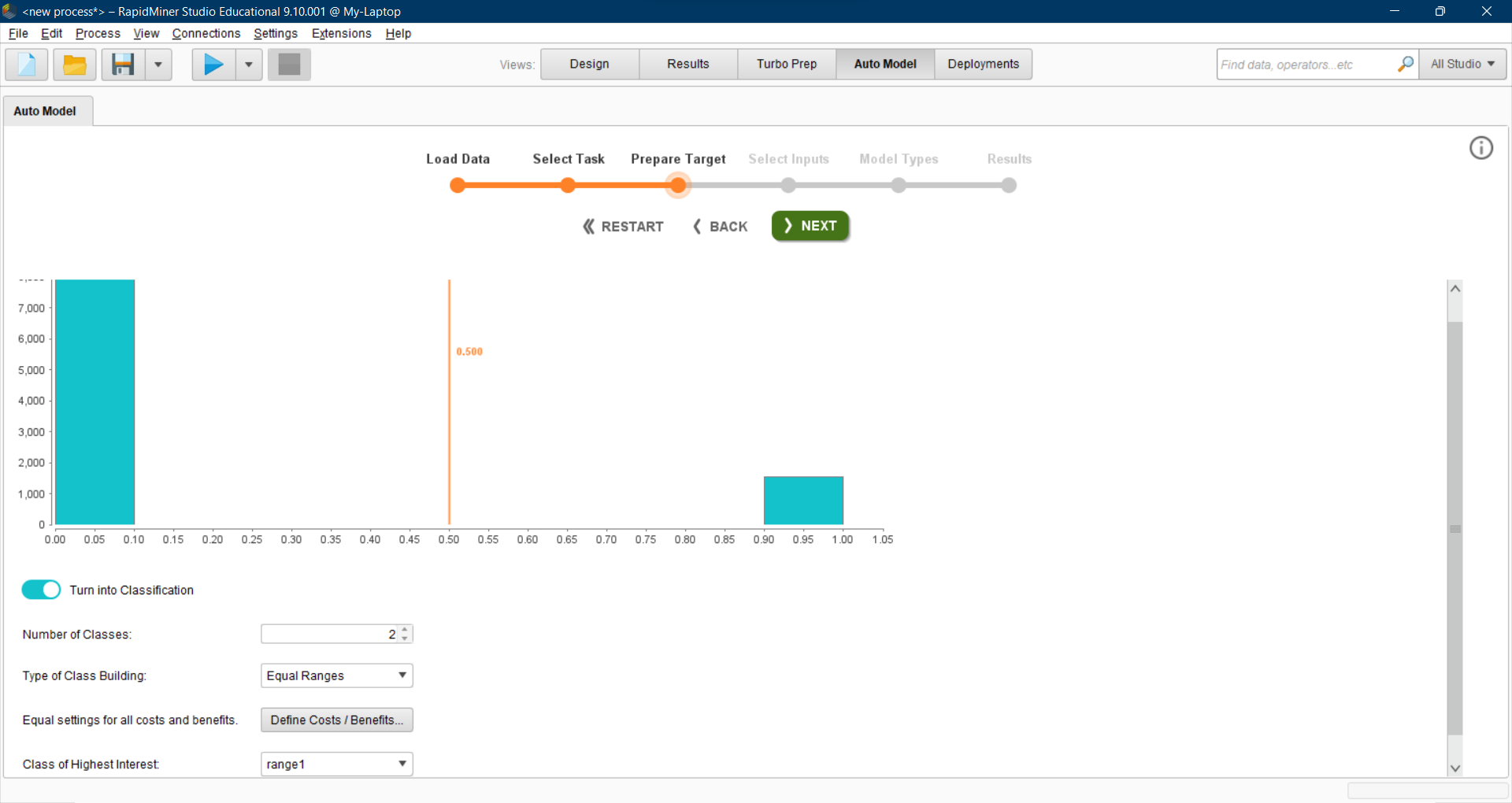


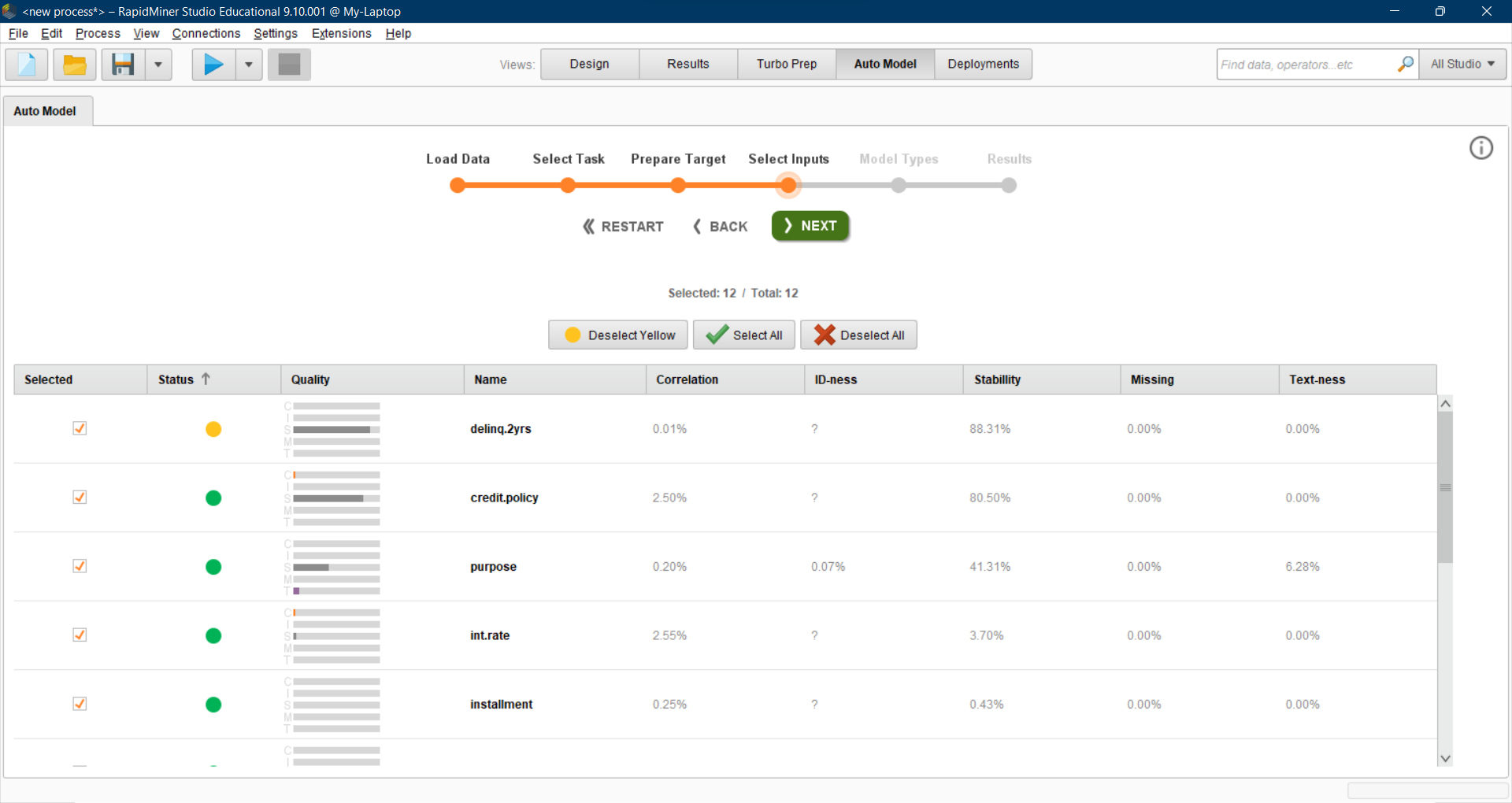


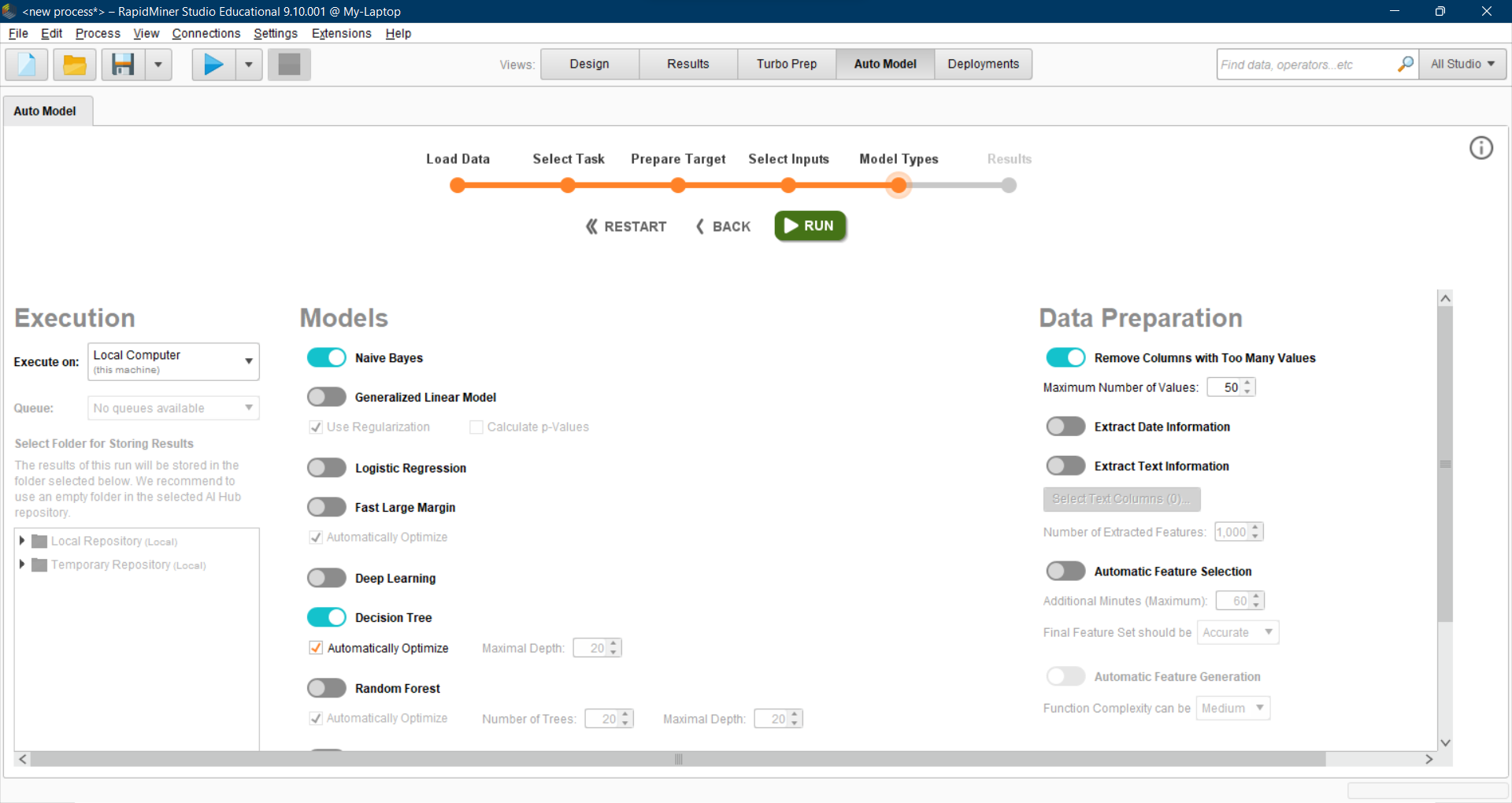
**Build Classification model using Rapid miner**











**Metrics and Comparison of Models**

