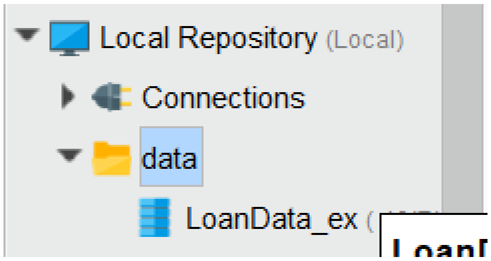
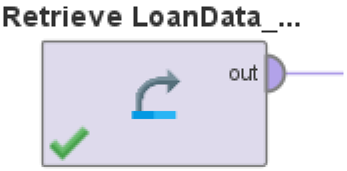
EX LAB 3

Create a neural network to predict risk levels for loan applicants at a bank.

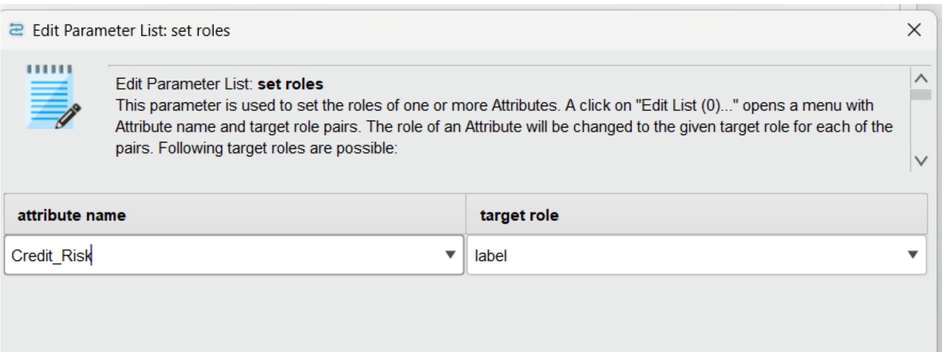
1) Download the training data set labeled *LoanData.csv.*



2) Import the training data set into your RapidMiner repository and name it descriptively. Drag and drop the data set into a new, blank main process.

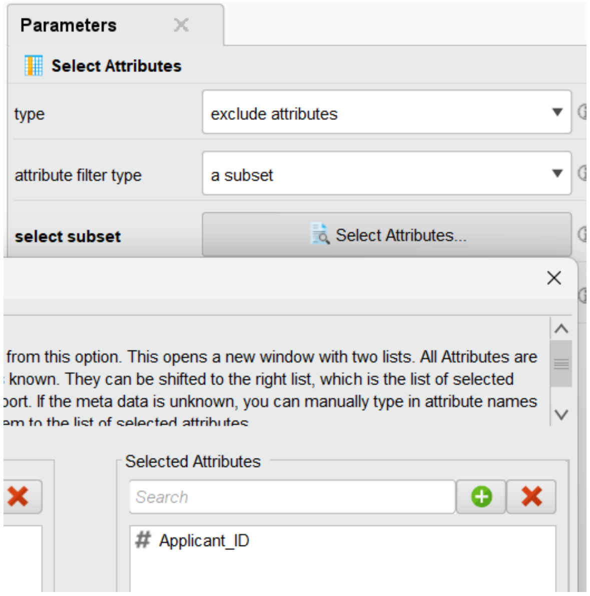


3) Set the Credit_Risk attribute as your label. Remember that Applicant_ID is not predictive.



Edit Parameter List: **set roles**
This parameter is used to set the roles of one or more Attributes. A click on "Edit List (0)..." opens a menu with Attribute name and target role pairs. The role of an Attribute will be changed to the given target role for each of the pairs. Following target roles are possible:

attribute name	target role
Credit_Risk	label



type: exclude attributes
attribute filter type: a subset
select subset

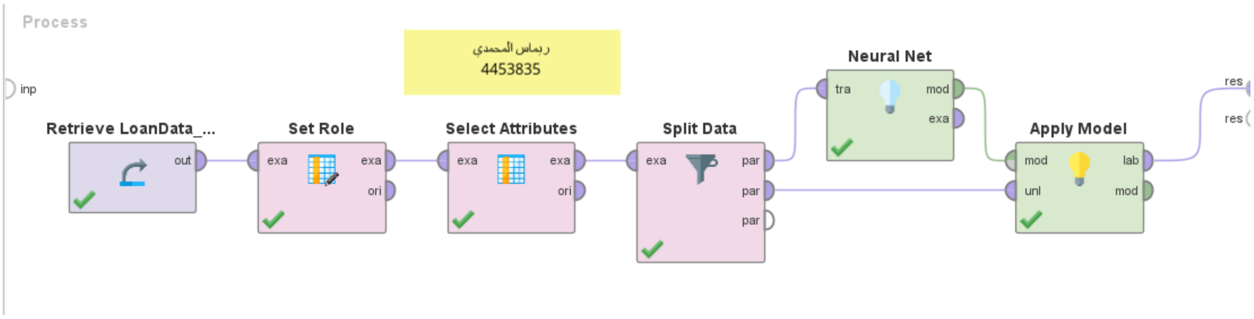
from this option. This opens a new window with two lists. All Attributes are
known. They can be shifted to the right list, which is the list of selected
port. If the meta data is unknown, you can manually type in attribute names
em to the list of selected attributes

Selected Attributes
Applicant_ID

I specified the column Credit_Risk that I want to predict and excluded the Applicant_ID because it is not important for prediction.

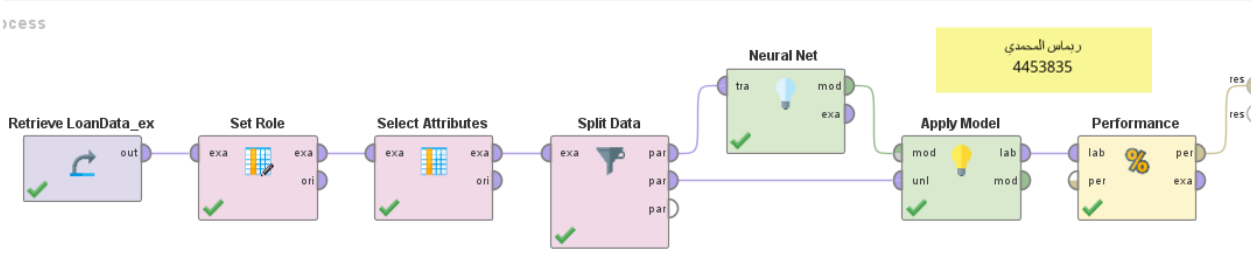
4) Add a Neural Net operator to your model.

5) You need another data set to apply the model. You can create another set of data or, use Split Data operator and use it on the current data that you have. Use Split Data operator for this practice.

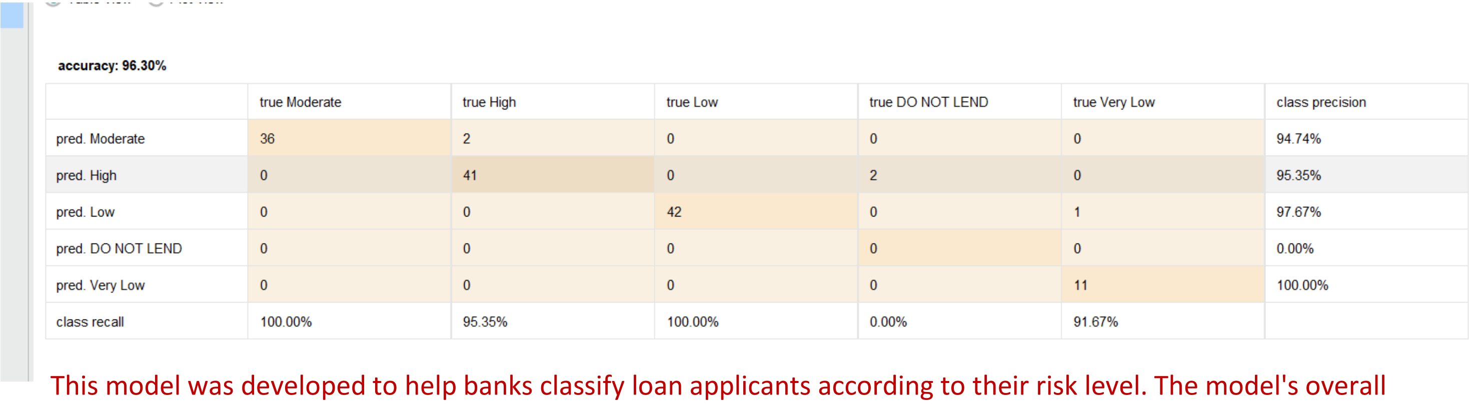


I trained the neural network model using the data. I divided the data into two parts: 0.7 for training and 0.3 for testing. I used the test portion to evaluate the model's performance.

6) Apply your model.



7) Run your model and review your predictions for each of your scoring observations. Report your results, including any interesting or unexpected results.



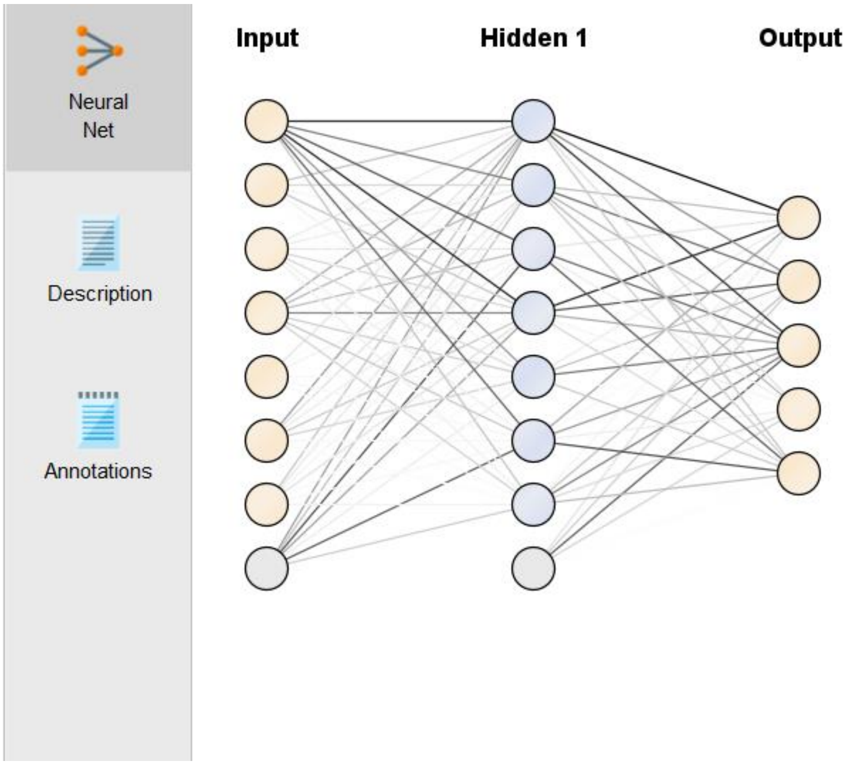
accuracy: 96.30%

	true Moderate	true High	true Low	true DO NOT LEND	true Very Low	class precision
pred. Moderate	36	2	0	0	0	94.74%
pred. High	0	41	0	2	0	95.35%
pred. Low	0	0	42	0	1	97.67%
pred. DO NOT LEND	0	0	0	0	0	0.00%
pred. Very Low	0	0	0	0	11	100.00%
class recall	100.00%	95.35%	100.00%	0.00%	91.67%	

This model was developed to help banks classify loan applicants according to their risk level. The model's overall accuracy was 96.3%, which means it's very successful at predicting risk. This is positive because it means the bank can easily determine whether a customer is high or low risk. However, the results are not equal for all categories. The model performed well in most categories, but it completely ignored the " DO NOT LEND " category.

This means that if a customer is very high risk or the bank has to reject them, the model will not predict them. This can cause decision-making problems, as the loan may be granted to a higher-risk individual.

The likely reason for this problem is that the number of customers who fall into the " DO NOT LEND" category in the training data is too small, or the model lacks the necessary data to differentiate between them. Alternatively, we need to improve the data or use a different algorithm that may be more effective.



Neural Network Components:

1. Input Layer:

Contains 8 nodes, equal to the number of features used from the data after removing the Applicant_ID.

2. Hidden Layer:

Responsible for processing the data and extracting hidden patterns and relationships.

3. Output Layer:

Contains 5 nodes, each representing a risk category:

DO NOT LEND

Very Low

Low

Moderate

High