Lecture 8
Modeling with Linear Regression

Kim, Yang Sok

Dept. of MIS, Keimyung University

Table of Content

Introduction

Introduction

- In this lecture, we also learn linear regression technique, a well-known prediction technique, accompanying with above issues

- In addition, we will focus on the following two issues
  - **Feature selection**: How to select **subset of attributes** that are appropriate for model learning?
  - **Parameter settings**: How to set appropriate parameters for model learning?

Linear Regression

https://machinelearningmastery.com/linear-regression-for-machine-learning/

- **Linear regression is a method for modeling the relationship between one or more independent variables and a dependent variable.**

X1, X2, …, Xn	→	Y
Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, …	⇨	quality

fixed acidity	volatile acidity	citric acid	residual sug...	chlorides	free sulfur d...	total sulfur d...	density	pH	sulphates	alcohol	quality
7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400 →	5
7.800	0.880	0	2.600	0.098	25	67	0.997	3.200	0.680	9.800 →	5
7.800	0.760	0.040	2.300	0.092	15	54	0.997	3.260	0.650	9.800 →	5
11.200	0.280	0.560	1.900	0.075	17	60	0.998	3.160	0.580	9.800 →	6
7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400 →	5

Linear Regression

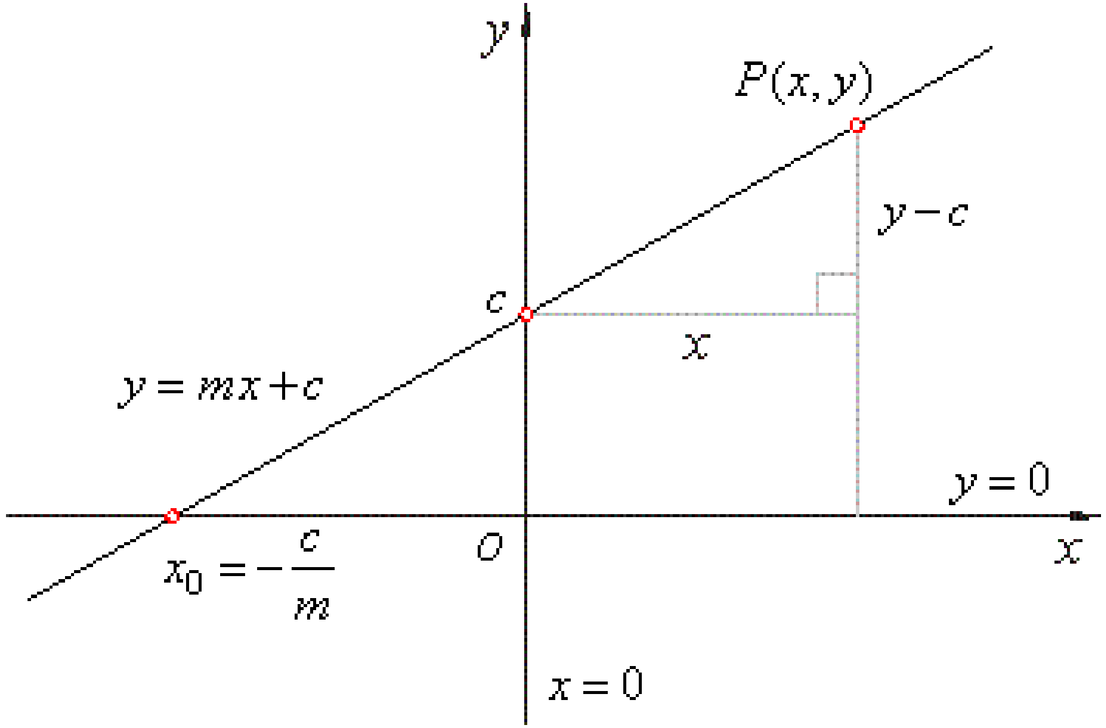
- **Let X be the independent variable and Y be the dependent variable. We will define a linear relationship between these two variables as follows:**

$$Y = mX + c$$

Intercept

Slope

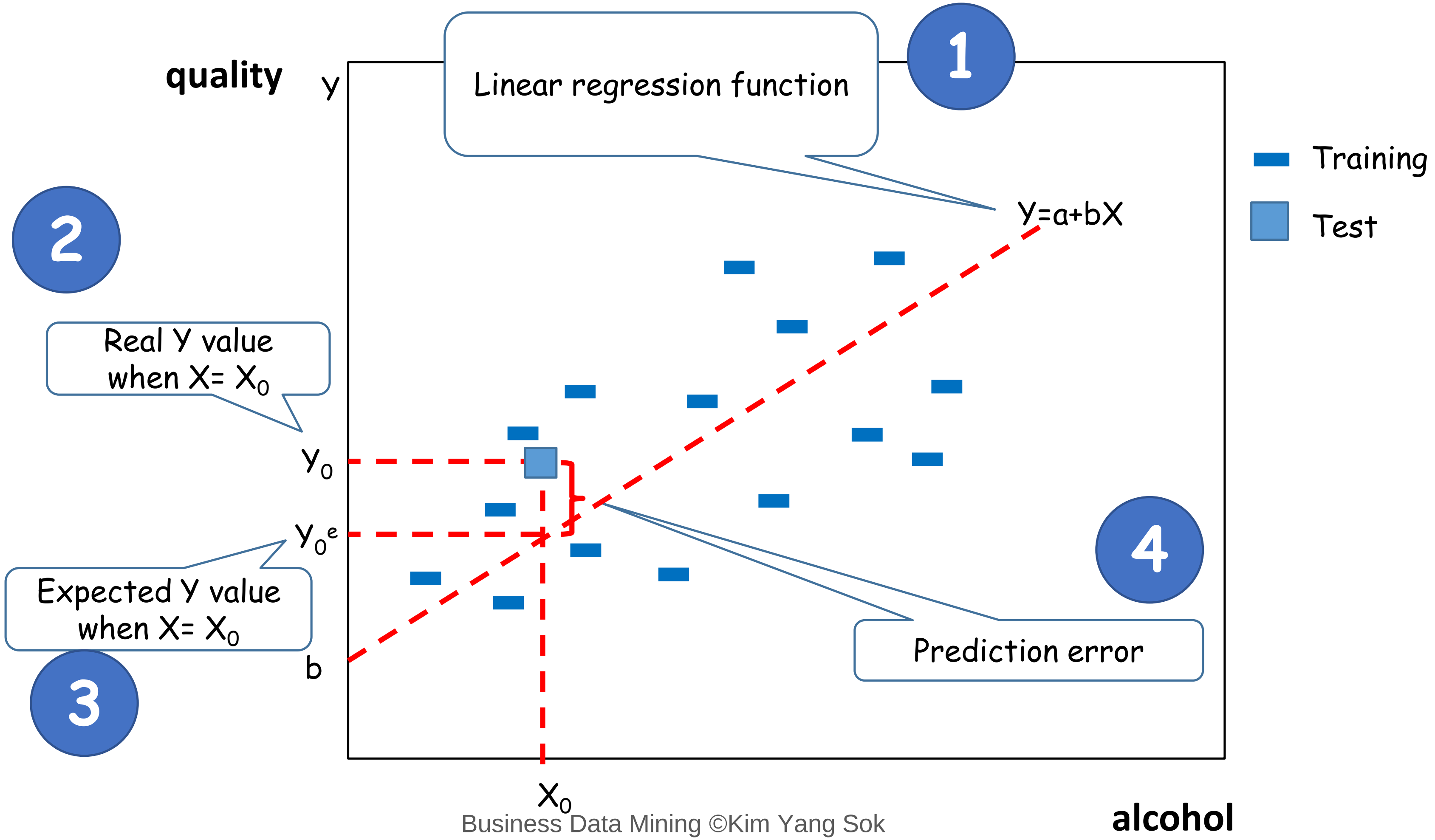
How to determine m and c?



Linear Regression

Concept

- **Linear regression is a method for modeling the relationship between one or more independent variables and a dependent variable.**



Linear Regression

Concept

- Linear regression was developed in the field of statistics, but has been borrowed by machine learning.
- Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).
- When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression.

- The loss is the error in our predicted value of m and c. Our goal is to minimize this error to obtain the most accurate value of m and c.
- We will use the **Mean Squared Error** function to calculate the loss. There are three steps in this function:

1. Find the difference between the actual y and predicted y value(y = mx + c), for a given x.
2. Square this difference.
3. Find the mean of the squares for every value in X.

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)^2$$

Linear Regression

How to Learn Linear Regression Model?

- **Here y_i is the actual value and $\bar{y}_i$ is the predicted value. Lets substitute the value of $\bar{y}_i$:**

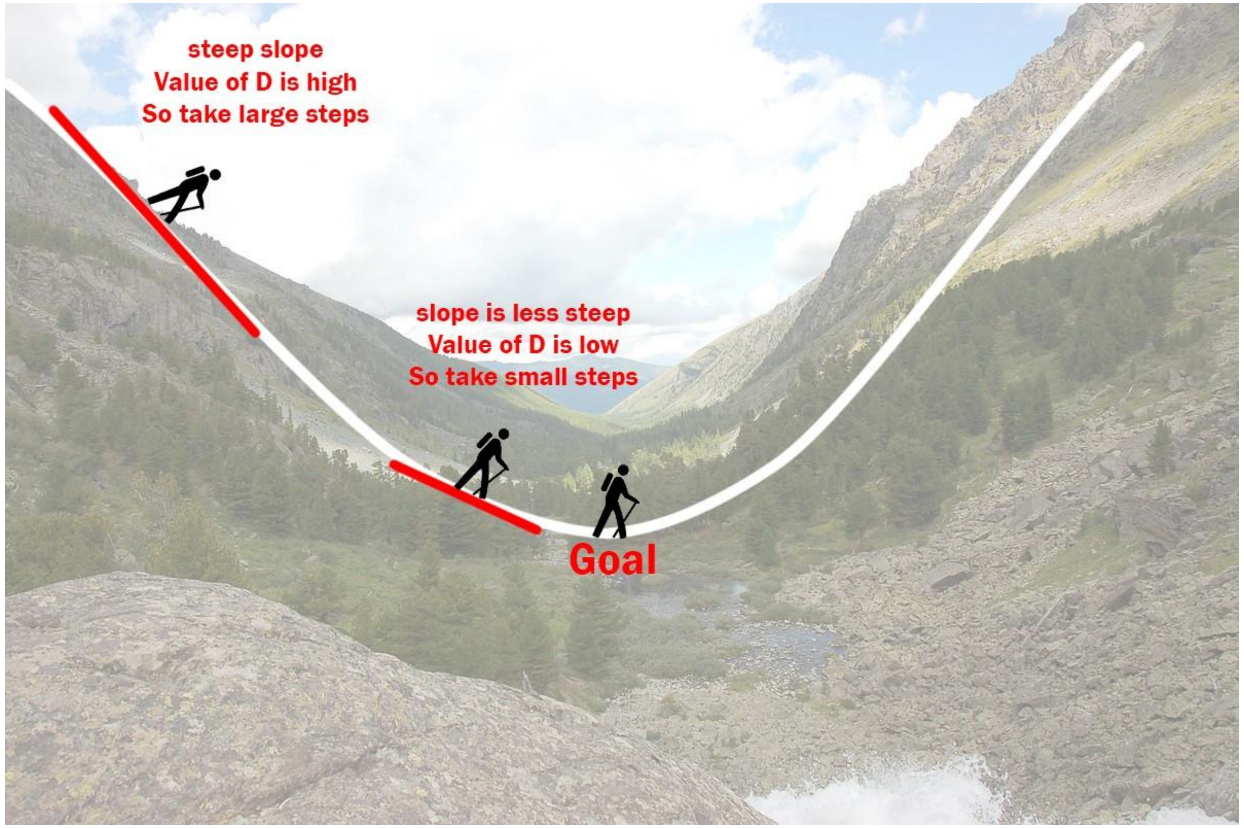
$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

- **So we square the error and find the mean. hence the name Mean Squared Error. Now that we have defined the loss function, lets get into the interesting part — minimizing it and finding m and c.**

Linear Regression

The Gradient Descent Algorithm

- **Gradient descent is an iterative optimization algorithm to find the minimum of a function. Here that function is our Loss Function.**
- **Understanding Gradient Descent**



Linear Regression

The Gradient Descent Algorithm

1. Initially let m = 0 and c = 0. Let L be our **learning rate**. This controls **how much the value of m changes with each step**. L could be a small value like 0.0001 for good accuracy.
2. Calculate **the partial derivative of the loss function** with respect to m, and plug in the current values of x, y, m and c in it to obtain the derivative value D

$$D_m = \frac{1}{n}\sum_{i=0}^{n} 2(y_i - (mx_i + c))(-x_i)$$

$$D_m = \frac{-2}{n}\sum_{i=0}^{n} x_i(y_i - \bar{y}_i)$$

D_m is the value of the partial derivative with respect to m. Similarly lets find the partial derivative with respect to c, Dc :

$$D_c = \frac{-2}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)$$

Linear Regression

The Gradient Descent Algorithm

3. Now we update the current value of **m** and **c** using the following equation:

$$m = m - L \times D_m$$

$$c = c - L \times D_c$$

4. We repeat this process until our loss function is a very small value or ideally 0 (which means 0 error or 100% accuracy). The value of **m** and **c** that we are left with now will be the optimum values.

Preparing Data For Linear Regression

- **Linear Assumption**
  - Linear regression assumes that the relationship between your input and output is linear. This may be obvious, but it is good to remember when you have a lot of attributes. You may need to transform data to make the relationship linear (e.g. log transform for an exponential relationship).
- **Remove Noise**
  - Linear regression assumes that your input and output variables are not noisy. Consider using data cleaning operations that let you better expose and clarify the signal in your data. This is most important for the output variable and you want to remove outliers in the output variable (y) if possible.
- **Remove Collinearity**
  - Linear regression will over-fit your data when you have highly correlated input variables. Consider calculating pairwise correlations for your input data and removing the most correlated.

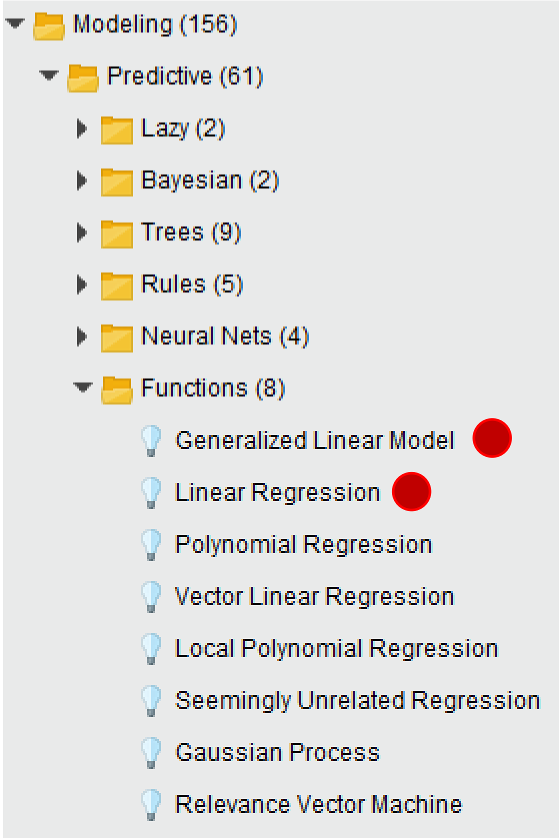
- **Gaussian Distributions**
  - Linear regression will make more reliable predictions if your input and output variables have a Gaussian distribution. You may get some benefit using transforms (e.g. log or BoxCox) on you variables to make their distribution more Gaussian looking.

- **Rescale Inputs**
  - Linear regression will often make more reliable predictions if you rescale input variables using standardization or normalization.

Linear Regression

Linear Regression in Rapidminer

- **Rapidminer provides <<Linear Regression>> in ‘Modeling>Predictive>Function’ package**



Exercise 1:
Linear Regression with Rapidminer

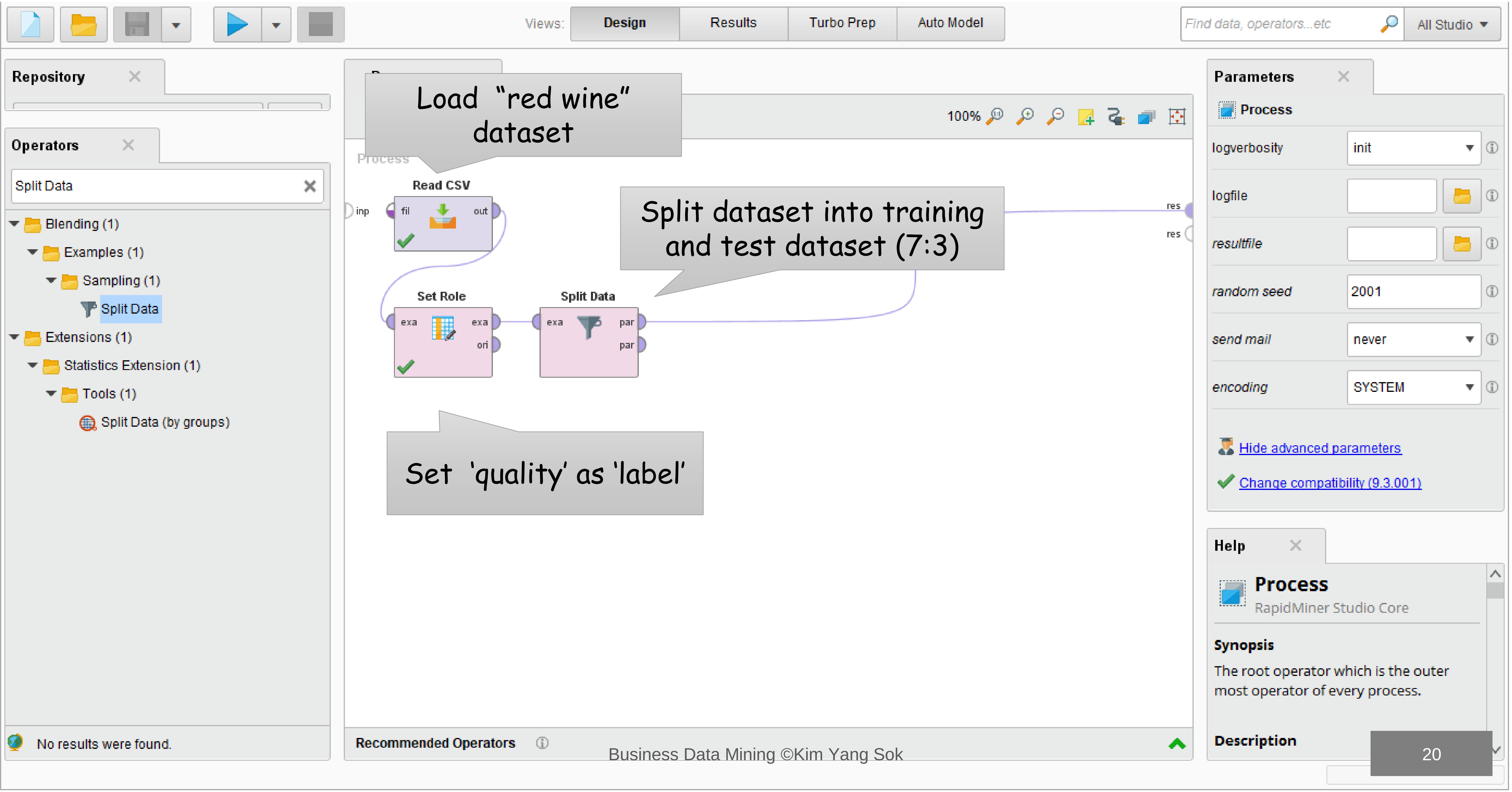
Exercise 1: Linear Regression with Rapidminer

Task & Process

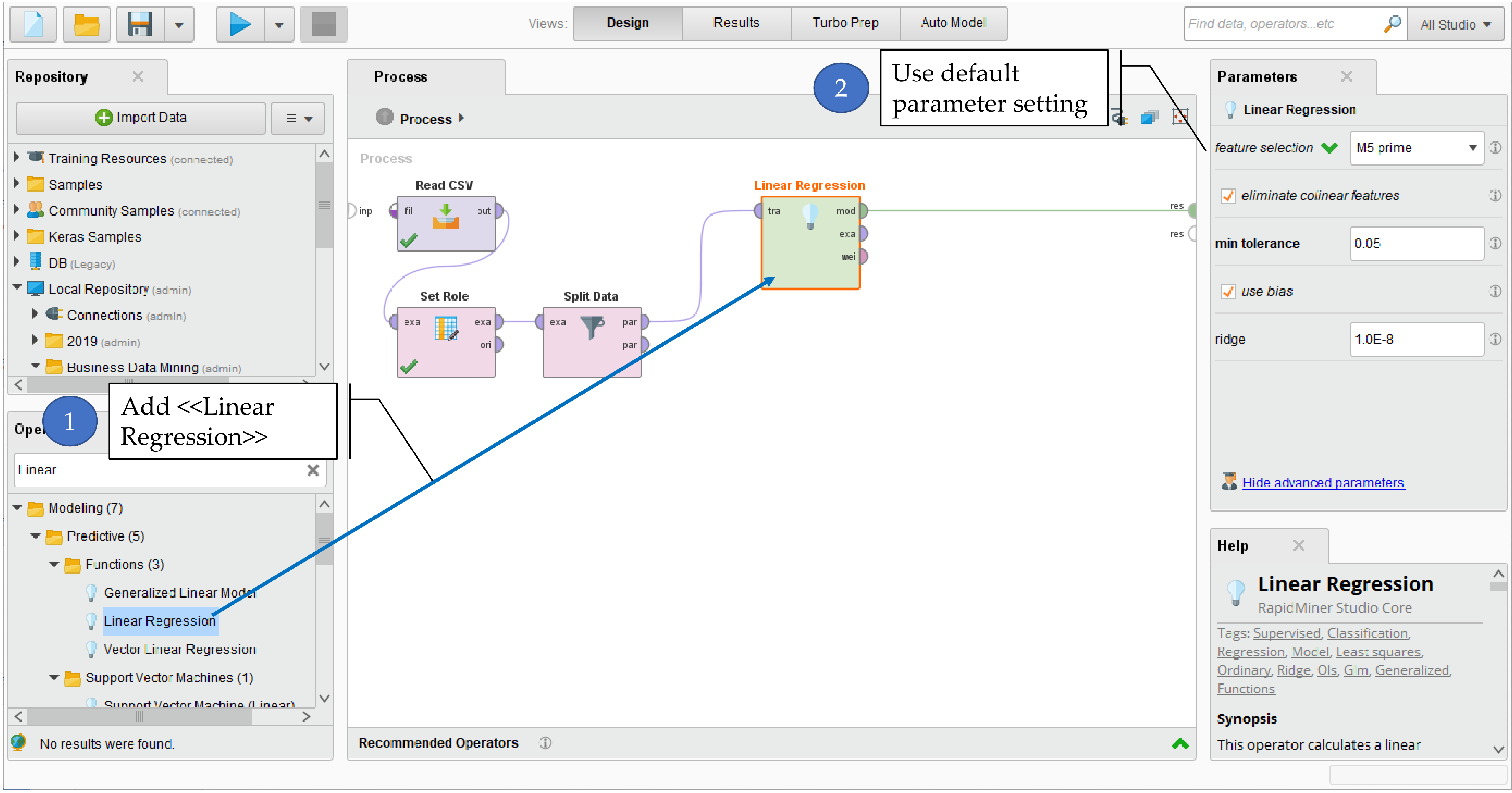
- **Task**
  - After loading data, build a linear regression model using **split test design**

- **Process**
  - Load "red wine" dataset
  - Set "quality" as label
  - Create split test design with <<Split Data>>
  - Create a linear regression model with the train dataset using linear regression algorithm
  - Apply the model to the test dataset
  - Set **regression** performance measures
  - Run the analysis process and check the performance results

Create split test design with <<Split Data>>



Create split test design with <<Split Data>>

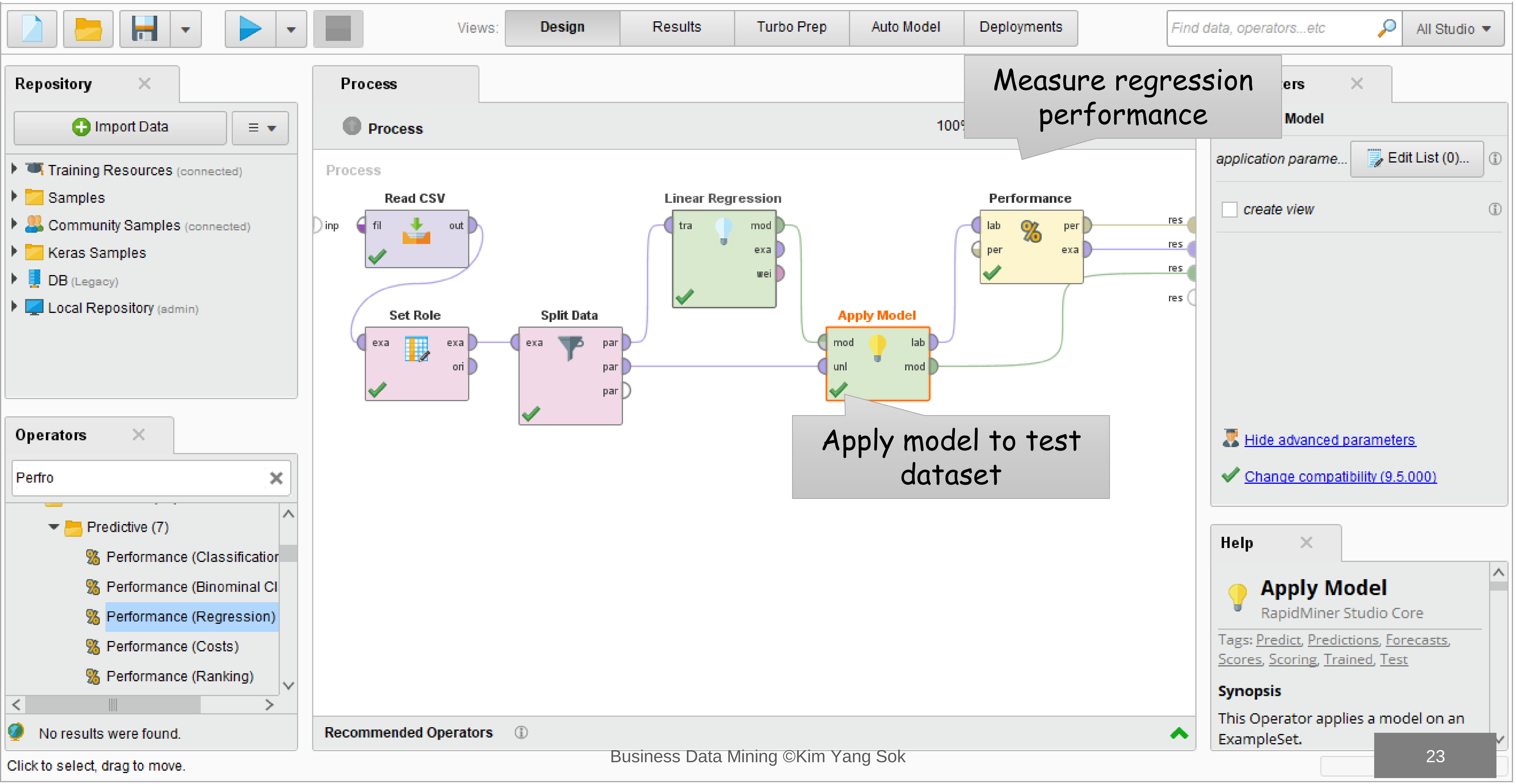


Exercise 1: Linear Regression with Rapidminer

Parameters of <<Linear Regression>>

- **feature selection**
  - Not all attributes are useful for linear regression.
  - Traditionally, forward selection (FS) or backward elimination (BE) are used to select features.
  - If you decide not to use these wrapper operators, you may use the ones which are bundled with multiple linear regression operator.
    - M5 prime, greedy, T-Test, iterative T-Test
- **eliminate collinear features**
  - This parameter indicates if the algorithm should try to delete collinear features during the regression or not.
  - min tolerance: This parameter is only available when the eliminate collinear features parameter is set to true. It specifies the minimum tolerance for eliminating collinear features.
- **use bias**
  - This parameter indicates if an intercept value should be calculated or not.
- **ridge**
  - This parameter specifies the ridge parameter for using in ridge regression.

Apply the model to the test dataset & Set regression performance measures



Run the analysis process and check the results - Model

Views: Design | Results | Turbo Prep | Auto Model | Deployments

Result History | LinearRegression (Linear Regression) | ExampleSet (Apply Model) | PerformanceVector (Performance)

Data

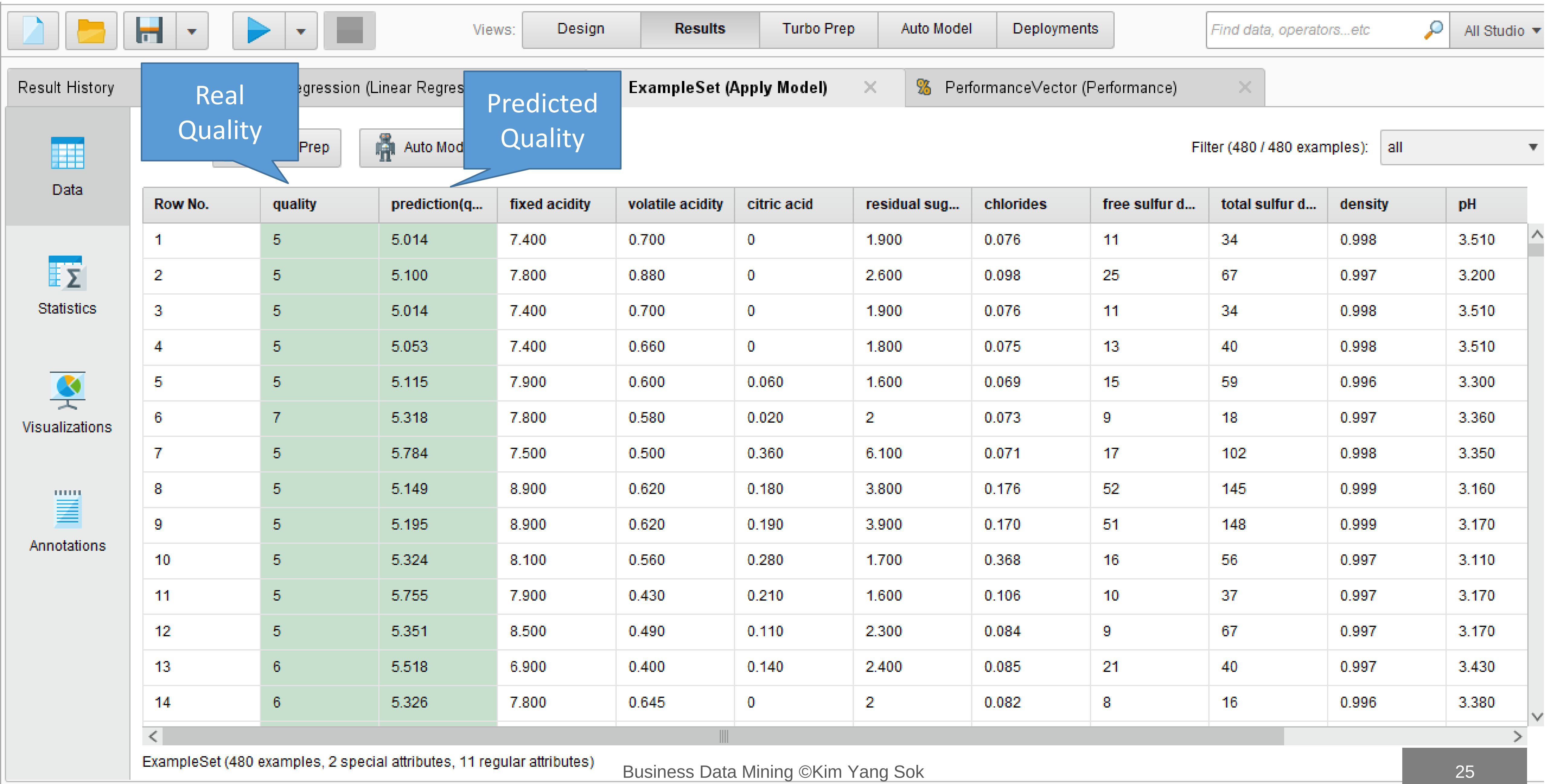
Description

Annotations

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
fixed acidity	0.018	0.020	0.038	0.956	0.921	0.357	
volatile acidity	-1.239	0.143	-0.269	0.826	-8.653	0	****
citric acid	-0.250	0.179	-0.060	0.815	-1.402	0.161	
residual sugar	0.021	0.015	0.037	1.000	1.471	0.142	
chlorides	-2.013	0.487	-0.122	0.994	-4.138	0.000	****
total sulfur dioxide	-0.002	0.001	-0.072	0.961	-2.714	0.007	***
pH	-0.520	0.186	-0.100	0.998	-2.800	0.005	***
sulphates	0.899	0.133	0.187	0.971	6.767	0.000	****
alcohol	0.301	0.021	0.396	0.883	14.474	0	****
(Intercept)	4.407	0.728	?	?	6.058	0.0	****

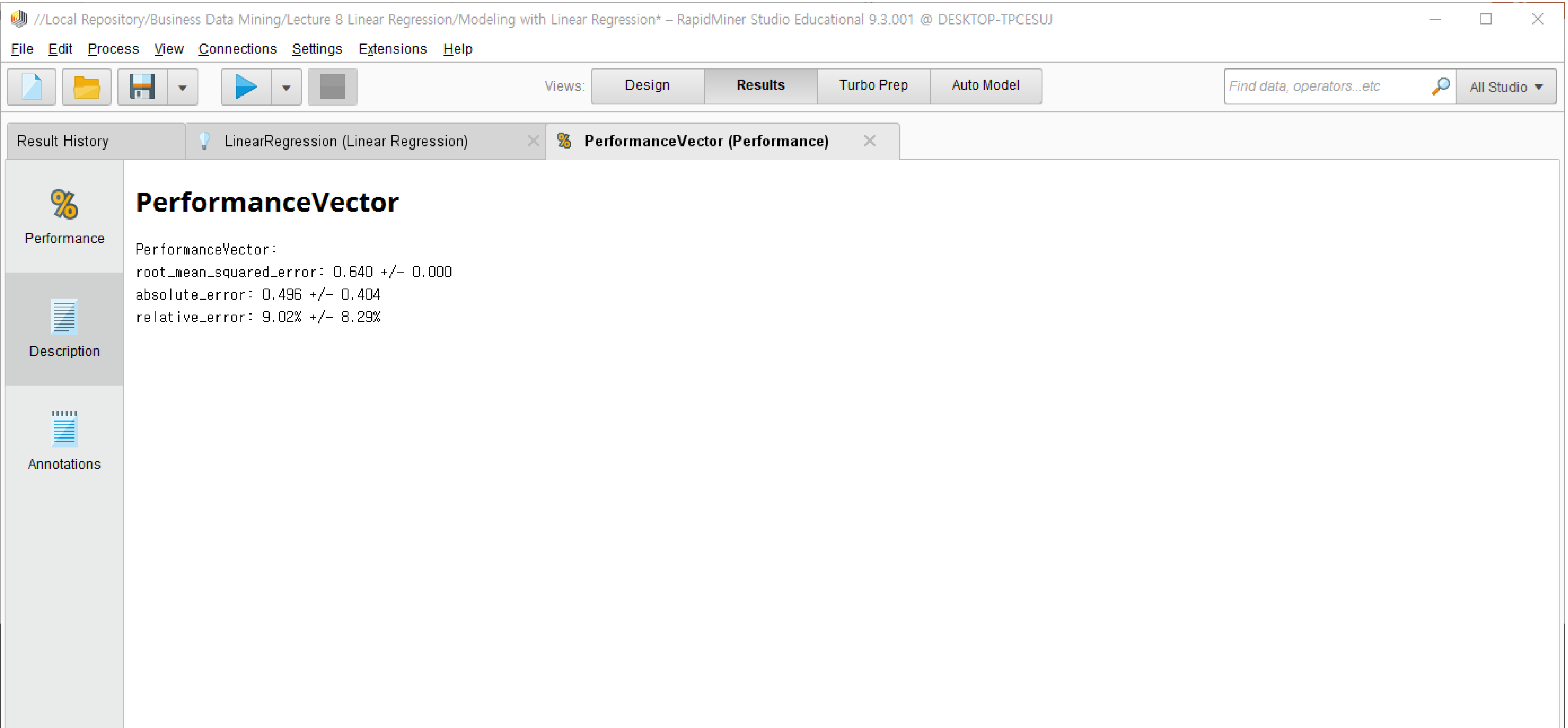
Check p-value and remove attribute p-value > 0.05

Run the analysis process and check the results - Examples



Row No.	quality	prediction(q...	fixed acidity	volatile acidity	citric acid	residual sug...	chlorides	free sulfur d...	total sulfur d...	density	pH
1	5	5.014	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510
2	5	5.100	7.800	0.880	0	2.600	0.098	25	67	0.997	3.200
3	5	5.014	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510
4	5	5.053	7.400	0.660	0	1.800	0.075	13	40	0.998	3.510
5	5	5.115	7.900	0.600	0.060	1.600	0.069	15	59	0.996	3.300
6	7	5.318	7.800	0.580	0.020	2	0.073	9	18	0.997	3.360
7	5	5.784	7.500	0.500	0.360	6.100	0.071	17	102	0.998	3.350
8	5	5.149	8.900	0.620	0.180	3.800	0.176	52	145	0.999	3.160
9	5	5.195	8.900	0.620	0.190	3.900	0.170	51	148	0.999	3.170
10	5	5.324	8.100	0.560	0.280	1.700	0.368	16	56	0.997	3.110
11	5	5.755	7.900	0.430	0.210	1.600	0.106	10	37	0.997	3.170
12	5	5.351	8.500	0.490	0.110	2.300	0.084	9	67	0.997	3.170
13	6	5.518	6.900	0.400	0.140	2.400	0.085	21	40	0.997	3.430
14	6	5.326	7.800	0.645	0	2	0.082	8	16	0.996	3.380

Run the analysis process and check the results - Performance



Exercise 2:
Linear Regression after Removing Insignificant Attributes

Run the analysis process and check the results - Model

Views: Design | Results | Turbo Prep | Auto Model | Deployments

Result History | LinearRegression (Linear Regression) | ExampleSet (Apply Model) | PerformanceVector (Performance)

Remove attributes with p-value > 0.05

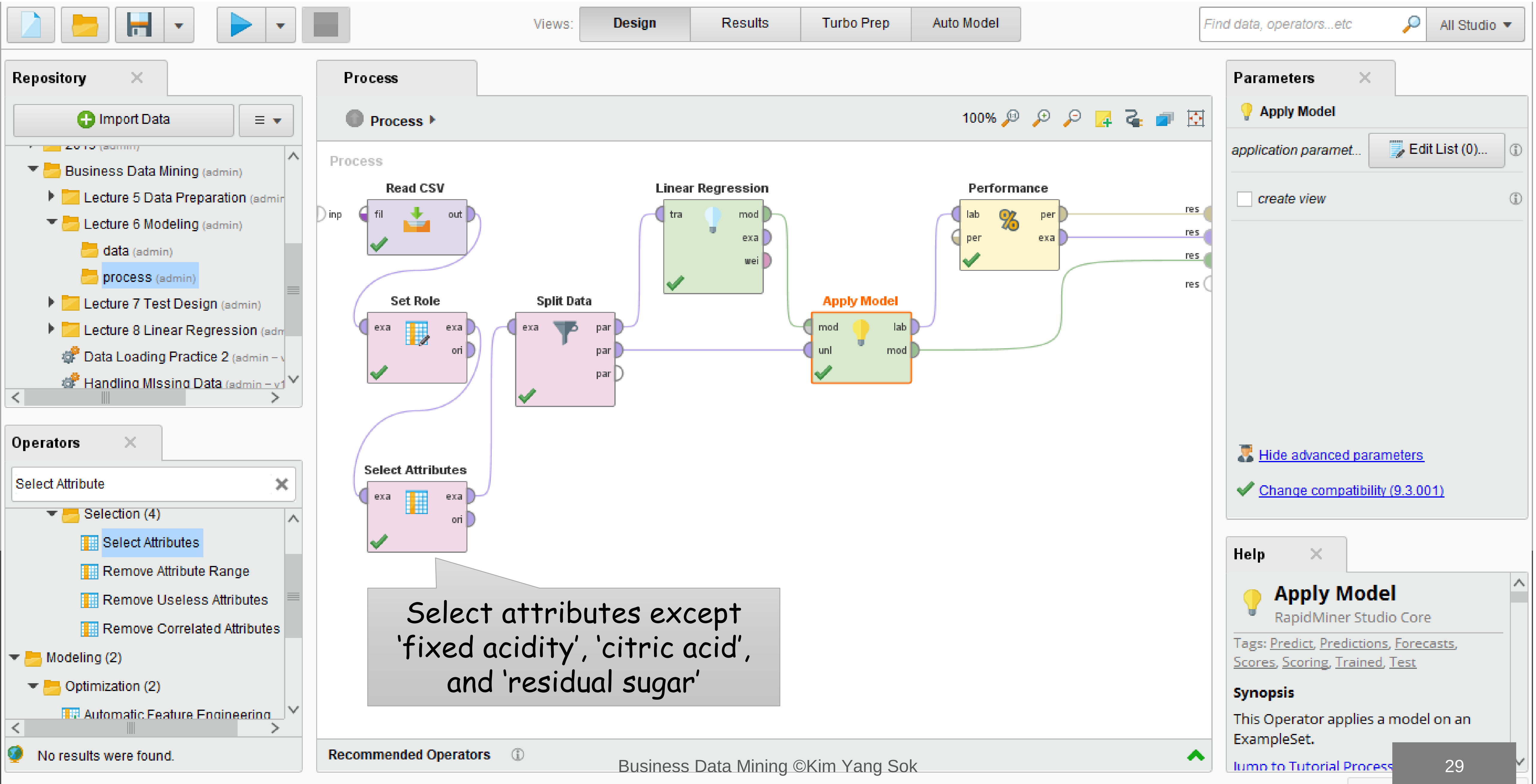
Data

Description

Annotations

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
fixed acidity	0.018	0.020	0.038	0.956	0.921	0.357	
volatile acidity	-1.239	0.143	-0.269	0.826	-8.653	0	****
citric acid	-0.250	0.179	-0.060	0.815	-1.402	0.161	
residual sugar	0.021	0.015	0.037	1.000	1.471	0.142	
chlorides	-2.013	0.487	-0.122	0.994	-4.138	0.000	****
total sulfur dioxide	-0.002	0.001	-0.072	0.961	-2.714	0.007	***
pH	-0.520	0.186	-0.100	0.998	-2.800	0.005	***
sulphates	0.899	0.133	0.187	0.971	6.767	0.000	****
alcohol	0.301	0.021	0.396	0.883	14.474	0	****
(Intercept)	4.407	0.728	?	?	6.058	0.000	****

Update process and run again!



Select attributes except 'fixed acidity', 'citric acid', and 'residual sugar'

Run the analysis process and check the results - Model

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
fixed acidity	0.018	0.020	0.038	0.956	0.921	0.357	
volatile acidity	-1.239	0.143	-0.269	0.826	-8.653	0	
citric acid	-0.250	0.179	-0.060	0.815	-1.402	0.161	
residual sugar	0.021	0.015	0.037	1.000	1.471	0.142	
chlorides	-2.013	0.487	-0.122	0.994	-4.138	0.000	****
total sulfur dioxide	-0.002	0.001	-0.072	0.961	-2.714	0.007	***
pH	-0.520	0.186	-0.100	0.998	-2.800	0.005	***
sulphates	0.899	0.133	0.187	0.971	6.767	0.000	****
alcohol	0.301	0.021	0.396	0.883	14.474	0	
(Intercept)	4.407	0.728	?	?	6.058	0.000	

PerformanceVector:
root_mean_squared_error: 0.640 +/- 0.000
absolute_error: 0.496 +/- 0.404
relative_error: 9.02% +/- 8.29%

PerformanceVector:
root_mean_squared_error: 0.638 +/- 0.000
absolute_error: 0.496 +/- 0.402
relative_error: 9.00% +/- 8.25%

Data
Description
Annotations

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
volatile acidity	-1.127	0.121	-0.244	0.880	-9.289	0	****
chlorides	-2.159	0.462	-0.130	0.997	-4.673	0.000	****
total sulfur dioxide	-0.002	0.001	-0.073	0.968	-2.977	0.003	***
pH	-0.548	0.140	-0.105	1.000	-3.924	0.000	****
sulphates	0.885	0.132	0.184	0.971	6.715	0.000	****
alcohol	0.298	0.020	0.392	0.902	14.790	0	****
(Intercept)	4.635	0.476	?	?	9.745	0	****

//Local Repository/Business Data Mining/Lecture 8 Linear Regression/Modeling with Linear Regression* – RapidMiner Studio Educational 9.3.001 @ DESKTOP-TPCESUJ

File Edit Process View Connections Settings Extensions Help

Views: Design | Results | Turbo Prep | Auto Model

Result History | LinearRegression (Linear Regression) | ExampleSet (Apply Model) | PerformanceVector (Performance)

Data | Statistics | Visualizations | Annotations

Open in Turbo Prep | Auto Model

Filter (480 / 480 examples): all

Row No.	quality	prediction(q...	volatile acidity	chlorides	free sulfur d...	total sulfur d...	density	pH	sulphates	alcohol
1	5	4.996	0.700	0.076	11	34	0.998	3.510	0.560	9.400
2	5	5.081	0.880	0.098	25	67	0.997	3.200	0.680	9.800
3	5	4.996	0.700	0.076	11	34	0.998		0.560	9.400
4	5	5.032	0.660	0.075	13	40	0.998		0.560	9.400
5	5	5.105	0.600	0.069	15	59	0.996	3.300	0.460	9.400
						40	0.997	3.430	0.630	9.700
						16	0.996	3.380	0.590	9.800
						113	0.997	3.170	0.660	9.800
16	5	5.340	0.320	0.103	13	50	0.996	3.380	0.550	9.200

Regression function

Quality = - 1.127 * volatile acidity
- 2.159 * chlorides
- 0.002 * total sulfur dioxide
- 0.548 * pH
+ 0.885 * sulphates
+ 0.298 * alcohol
+ 4.635

	coefficient	attribute value	multiplication
volatile acidity	-1.127	0.7	-0.789
chlorides	-2.159	0.076	-0.164
total sulfur dioxide	-0.002	34	-0.068
pH	-0.548	3.51	-1.923
sulphates	0.885	0.56	3.106
alcohol	0.298	9.4	0.167
Intercept			4.635
		prediction	**4.964**

ExampleSet (480 examples, 2 special attributes, 8 regular attributes)

Exercise 3: Linear Regression with Variance

Feature Selection

Feature Selection with Variance Thresholds

- Variance thresholds remove features whose values don't change much from observation to observation (i.e. their variance falls below a threshold). These features provide little value.

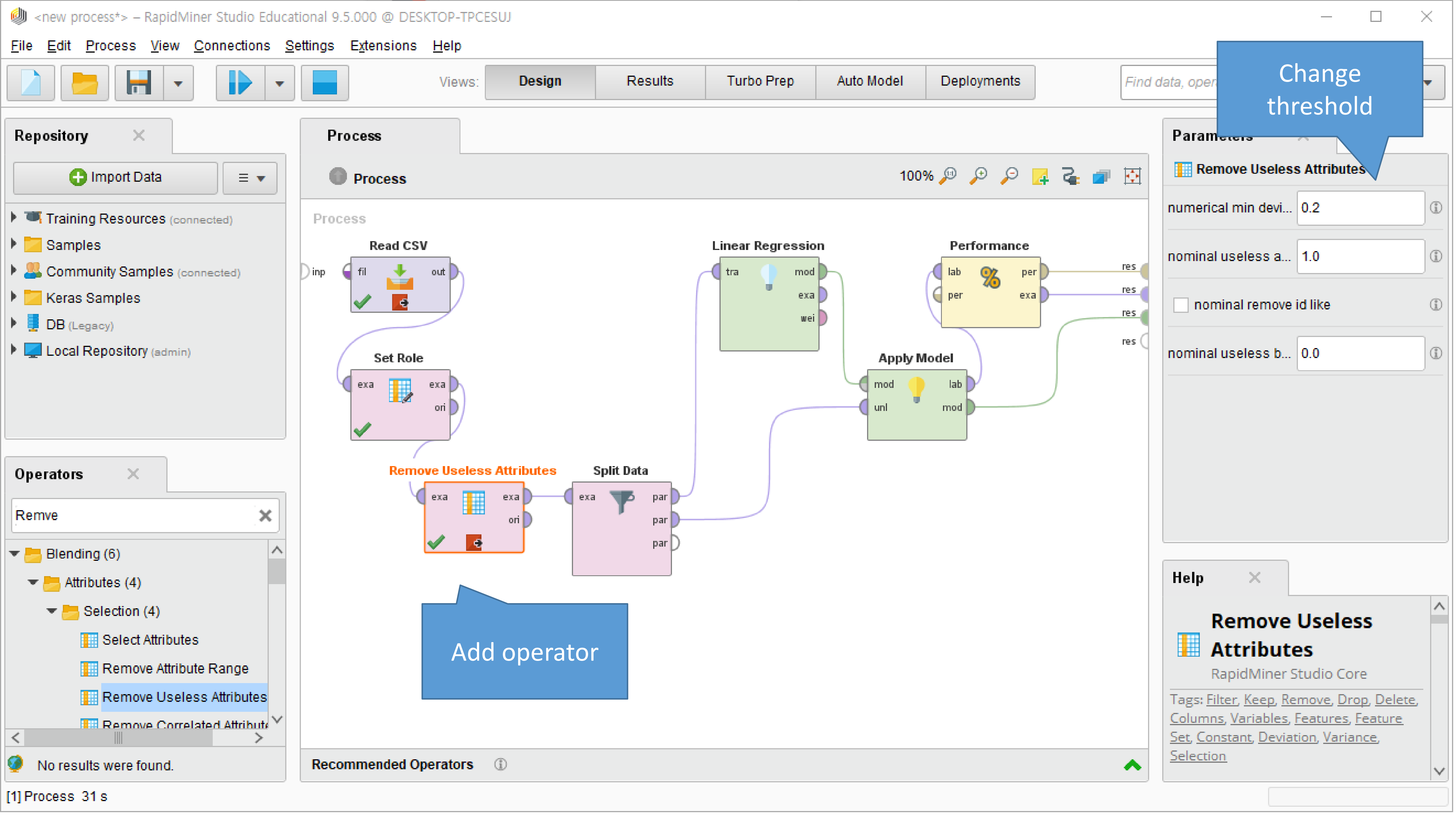
- For example, if you had a public health dataset where 96% of observations were for 35-year-old men, then the 'Age' and 'Gender' features can be eliminated without a major loss in information.

- Because variance is dependent on scale, you should always normalize your features first.

Feature Selection with Variance Thresholds

- Strengths
  - Applying variance thresholds is based on solid intuition: features that don't change much also don't add much information.
  - This is an easy and relatively safe way to reduce dimensionality at the start of your modeling process.

- Weaknesses
  - If your problem does require dimensionality reduction, applying variance thresholds is rarely sufficient.
  - Furthermore, you must manually set or tune a variance threshold, which could be tricky.
  - It is better to start with a conservative (i.e. lower) threshold.


<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, oper
Change threshold
Repository
Import Data
Training Resources (connected)
Samples
Community Samples (connected)
Keras Samples
DB (Legacy)
Local Repository (admin)
Operators
Remve
Blending (6)
Attributes (4)
Selection (4)
Select Attributes
Remove Attribute Range
Remove Useless Attributes
Remove Correlated Attribut
No results were found.
Process
Process
100%
Process
Read CSV
Set Role
Remove Useless Attributes
Split Data
Linear Regression
Apply Model
Performance
Add operator
Recommended Operators
Parameters
Remove Useless Attributes
numerical min devi... 0.2
nominal useless a... 1.0
nominal remove id like
nominal useless b... 0.0
Help
Remove Useless Attributes
RapidMiner Studio Core
Tags: Filter, Keep, Remove, Drop, Delete, Columns, Variables, Features, Feature Set, Constant, Deviation, Variance, Selection
[1] Process 31 s

Exercise 4: Linear Regression with Correlation

Feature Selection with Correlation Thresholds

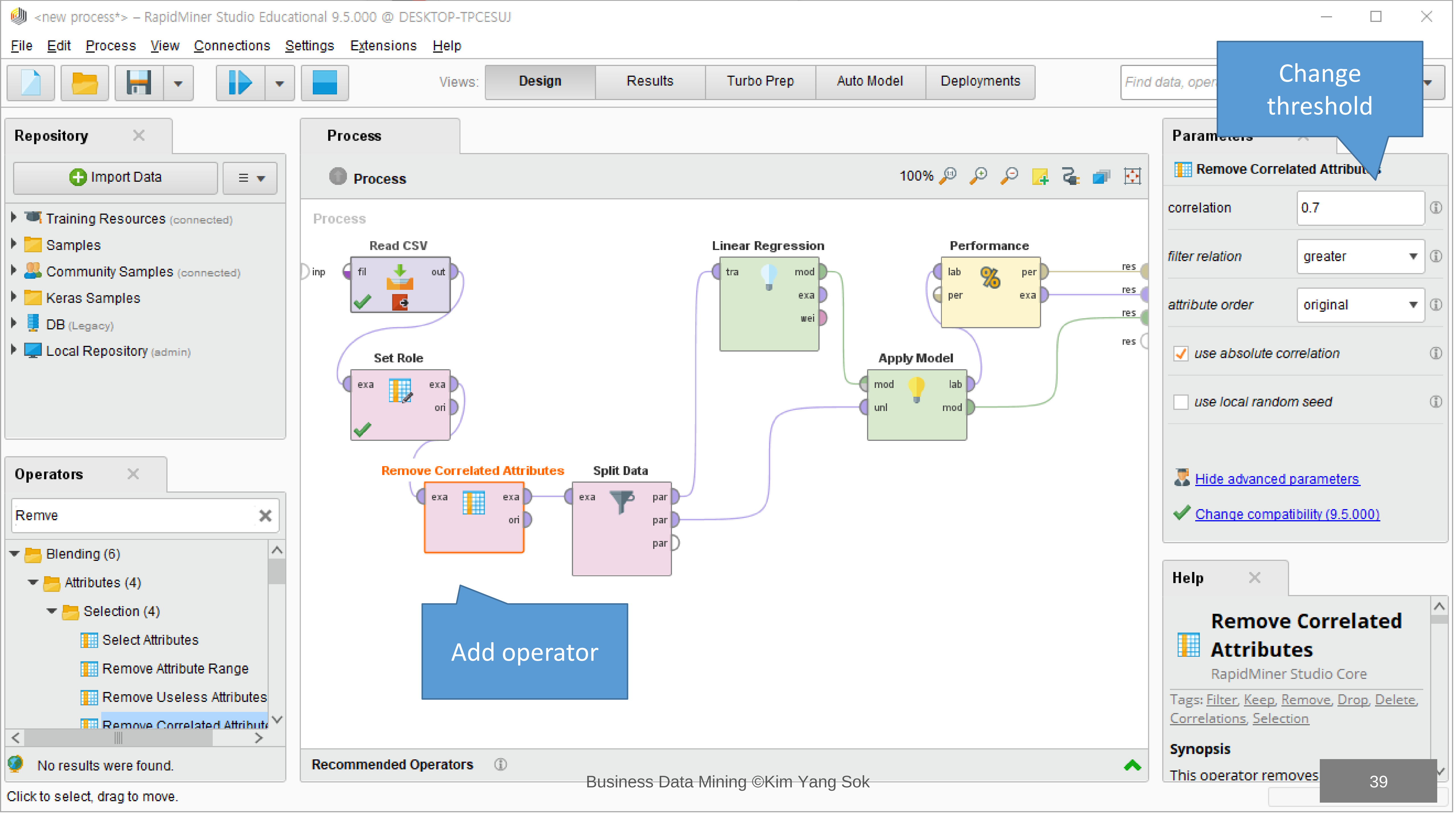
- Correlation thresholds remove features that are highly correlated with others (i.e. its values change very similarly to another's). These features provide redundant information.

- For example, if you had a real-estate dataset with 'Floor Area (Sq. Ft.)' and 'Floor Area (Sq. Meters)' as separate features, you can safely remove one of them.

- Which one should you remove? Well, you'd first calculate all pair-wise correlations. Then, if the correlation between a pair of features is above a given threshold, you'd remove the one that has larger mean absolute correlation with other features.

- Strengths
  - Applying correlation thresholds is also based on solid intuition: similar features provide redundant information.
  - Some algorithms are not robust to correlated features, so removing them can boost performance.

- Weaknesses:
  - You must manually set or tune a correlation threshold, which can be tricky to do.
  - In addition, if you set your threshold too low, you risk dropping useful information.
  - Whenever possible, we prefer algorithms with built-in feature selection over correlation thresholds.
  - Even for algorithms without built-in feature selection, Principal Component Analysis (PCA) is often a better alternative.



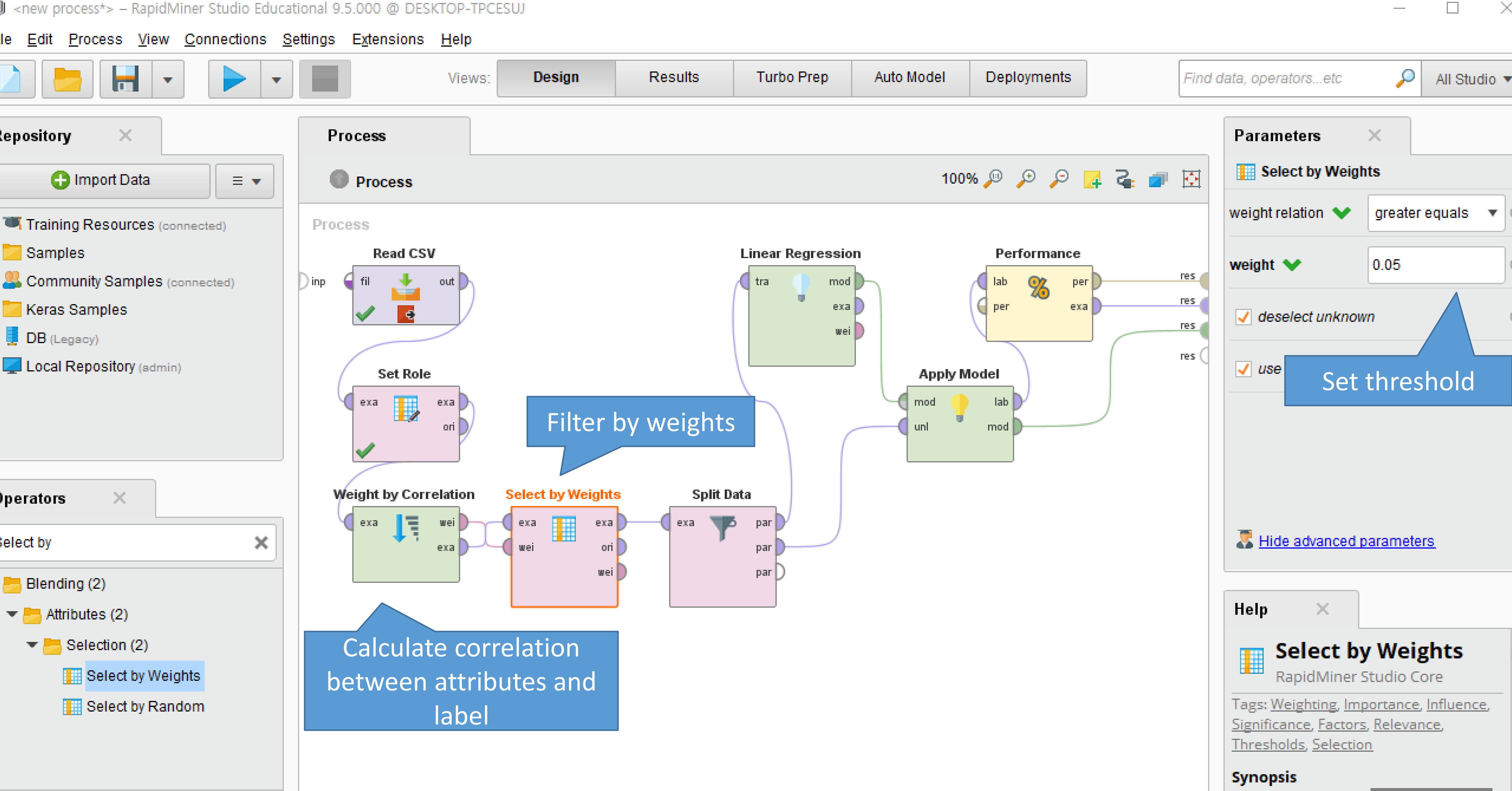
<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Change threshold
Repository
Import Data
Training Resources (connected)
Samples
Community Samples (connected)
Keras Samples
DB (Legacy)
Local Repository (admin)
Process
100%
Read CSV
Set Role
Linear Regression
Performance
Apply Model
Remove Correlated Attributes
Split Data
Add operator
Parameters
Remove Correlated Attributes
correlation
0.7
filter relation
greater
attribute order
original
use absolute correlation
use local random seed
Hide advanced parameters
Change compatibility (9.5.000)
Operators
Remve
Blending (6)
Attributes (4)
Selection (4)
Select Attributes
Remove Attribute Range
Remove Useless Attributes
Remove Correlated Attribute
No results were found.
Click to select, drag to move.
Recommended Operators
Help
Remove Correlated Attributes
RapidMiner Studio Core
Tags: Filter, Keep, Remove, Drop, Delete, Correlations, Selection
Synopsis
This operator removes

Exercise 5: Linear Regression with Weight

Feature Selection

Feature Selection with Filter Methods

- Filter feature selection methods apply a statistical measure to assign a scoring to each feature.
- The features are ranked by the score and either selected to be kept or removed from the dataset.
- The methods are often univariate and consider the feature independently, or with regard to the dependent variable.
- Some examples of some filter methods include the Chi squared test, information gain and correlation coefficient scores.


Read CSV
Set Role
Weight by Correlation
Select by Weights
Split Data
Linear Regression
Apply Model
Performance
Filter by weights
Calculate correlation between attributes and label
Set threshold
weight relation
greater equals
weight
0.05
deselect unknown

Exercise 6:
Linear Regression with Forward Selection using Rapidminer

Feature Selection with Stepwise Search

- Stepwise search is a supervised feature selection method based on sequential search, and it has two flavors: forward and backward.
- Forward stepwise search
  - Start without any features.
  - Then, you'd train a 1-feature model using each of your candidate features and keep the version with the best performance.
  - You'd continue adding features, one at a time, until your performance improvements stall.
- Backward stepwise search
  - Start with all features in your model and then remove one at a time until performance starts to drop substantially.
- Despite many textbooks listing stepwise search as a valid option, it almost always underperforms other supervised methods such as regularization.

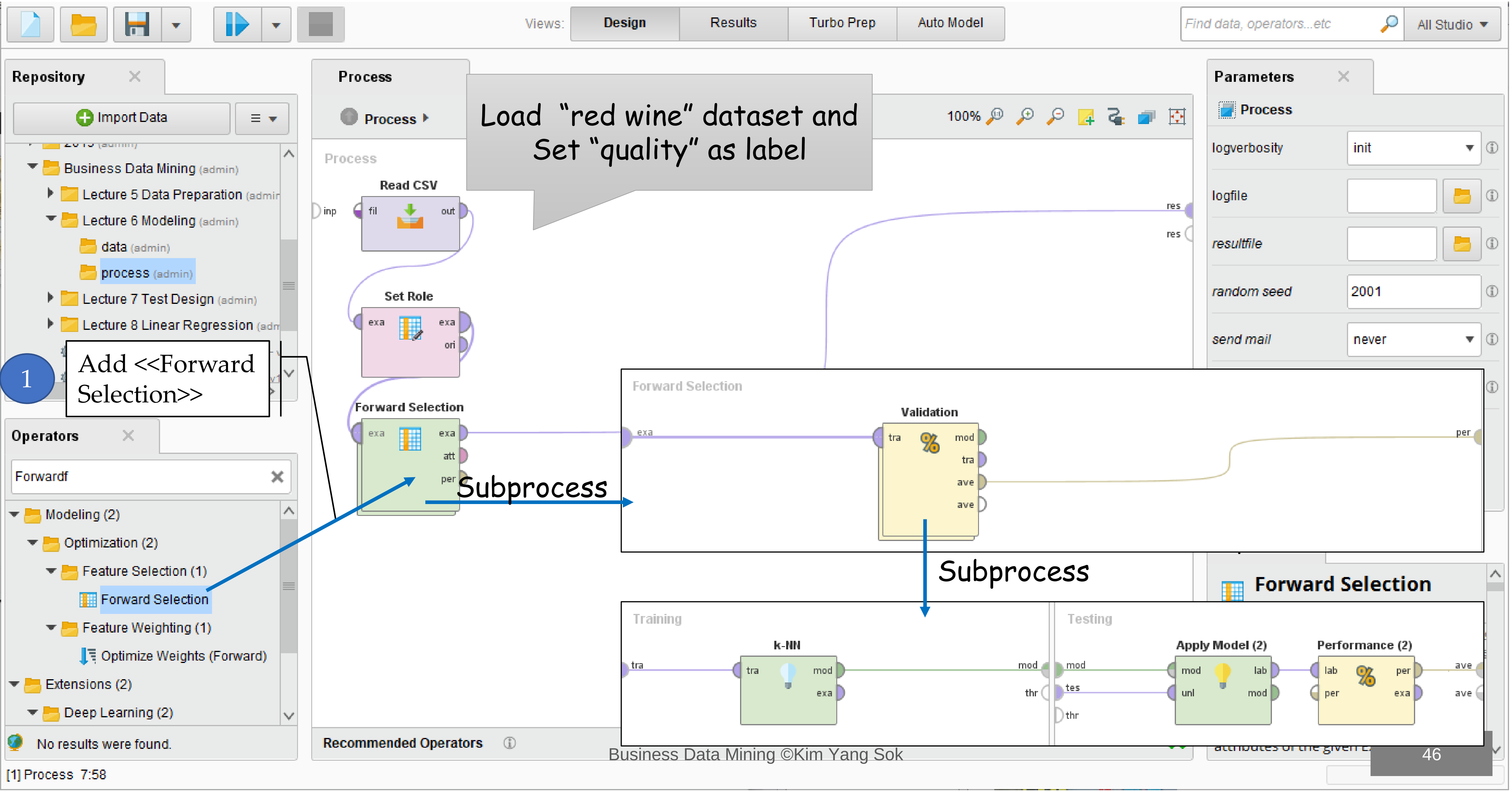
Exercise 6: Linear Regression with Feature Selection using Rapidminer

Task & Process

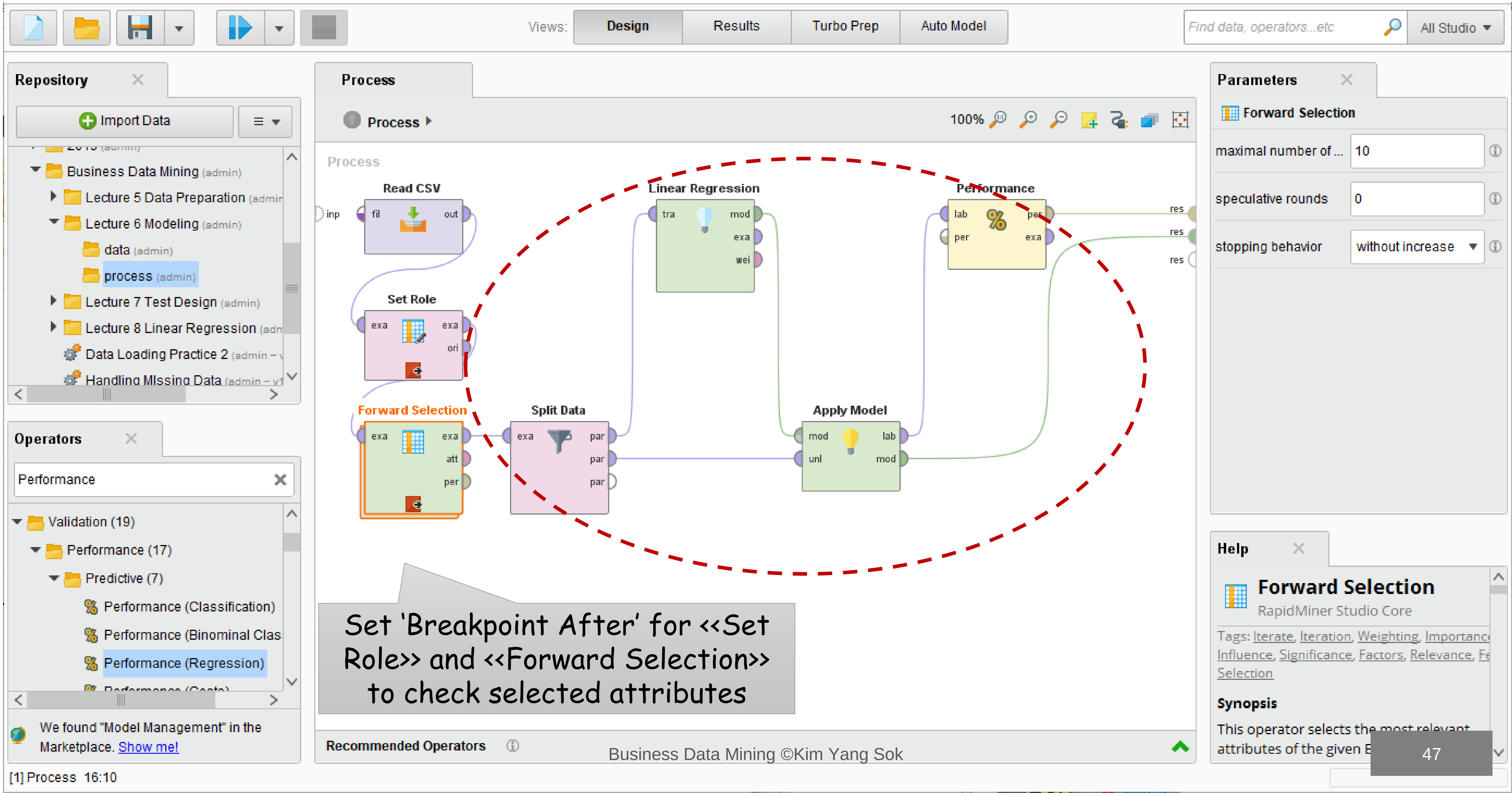
- **Task**
  - After loading data, build a linear regression model with feature selection.

- **Process**
  - Load “red wine” dataset and Set “quality” as label
  - Select attributes with forward selection approach
  - Perform a linear regression modeling with the split test design
  - Run the analysis process and evaluate analysis results

Select attributes with forward selection approach

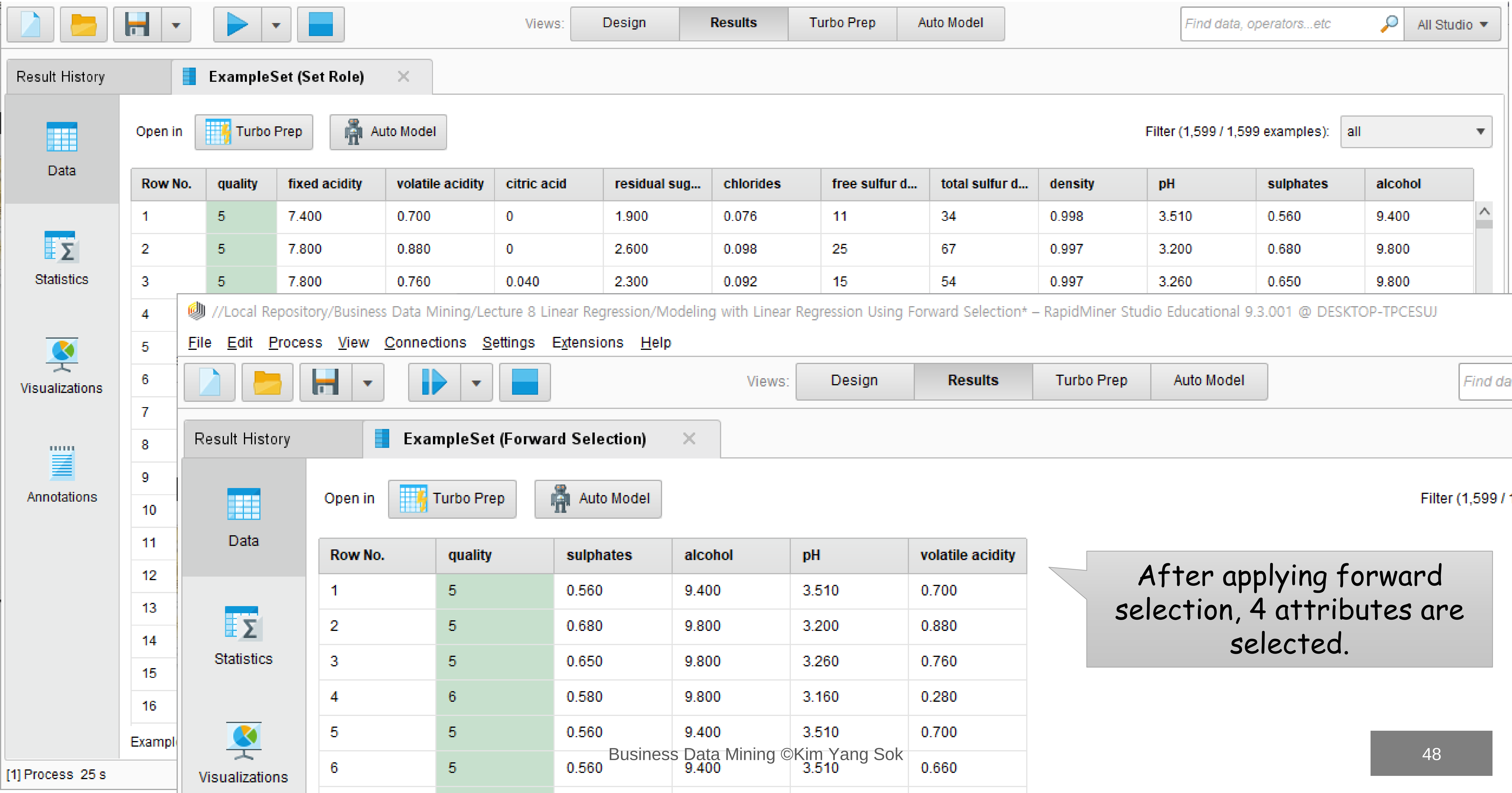


Perform a linear regression modeling with the split test design



Set 'Breakpoint After' for <<Set Role>> and <<Forward Selection>> to check selected attributes

Run the analysis process and evaluate analysis results



Row No.	quality	fixed acidity	volatile acidity	citric acid	residual sug...	chlorides	free sulfur d...	total sulfur d...	density	pH	sulphates	alcohol
1	5	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400
2	5	7.800	0.880	0	2.600	0.098	25	67	0.997	3.200	0.680	9.800
3	5	7.800	0.760	0.040	2.300	0.092	15	54	0.997	3.260	0.650	9.800
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Row No.	quality	sulphates	alcohol	pH	volatile acidity
1	5	0.560	9.400	3.510	0.700
2	5	0.680	9.800	3.200	0.880
3	5	0.650	9.800	3.260	0.760
4	6	0.580	9.800	3.160	0.280
5	5	0.560	9.400	3.510	0.700
6	5	0.560	9.400	3.510	0.660

After applying forward selection, 4 attributes are selected.

Run the analysis process and evaluate analysis results

Views: Design | Results | Turbo Prep | Auto Model

Result History | LinearRegression (Linear Regression) | PerformanceVector (Performance)

Data

Description

Annotations

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
sulphates	0.560	0.115	0.124	0.945	4.868	0.000	****
alcohol	0.342	0.020	0.439	0.959	17.156	0	****
pH	-0.392	0.136	-0.076	1.000	-2.880	0.004	***
volatile acidity	-1.251	0.115	-0.282	0.910	-10.843	0	****
(Intercept)	3.665	0.449	?	?	8.167	0.000	****

```
PerformanceVector:
root_mean_squared_error: 0.693 +/- 0.000
absolute_error: 0.540 +/- 0.434
relative_error: 9.93% +/- 9.50%
```

Exercise 7: Linear Regression with Backward Elimination using Rapidminer

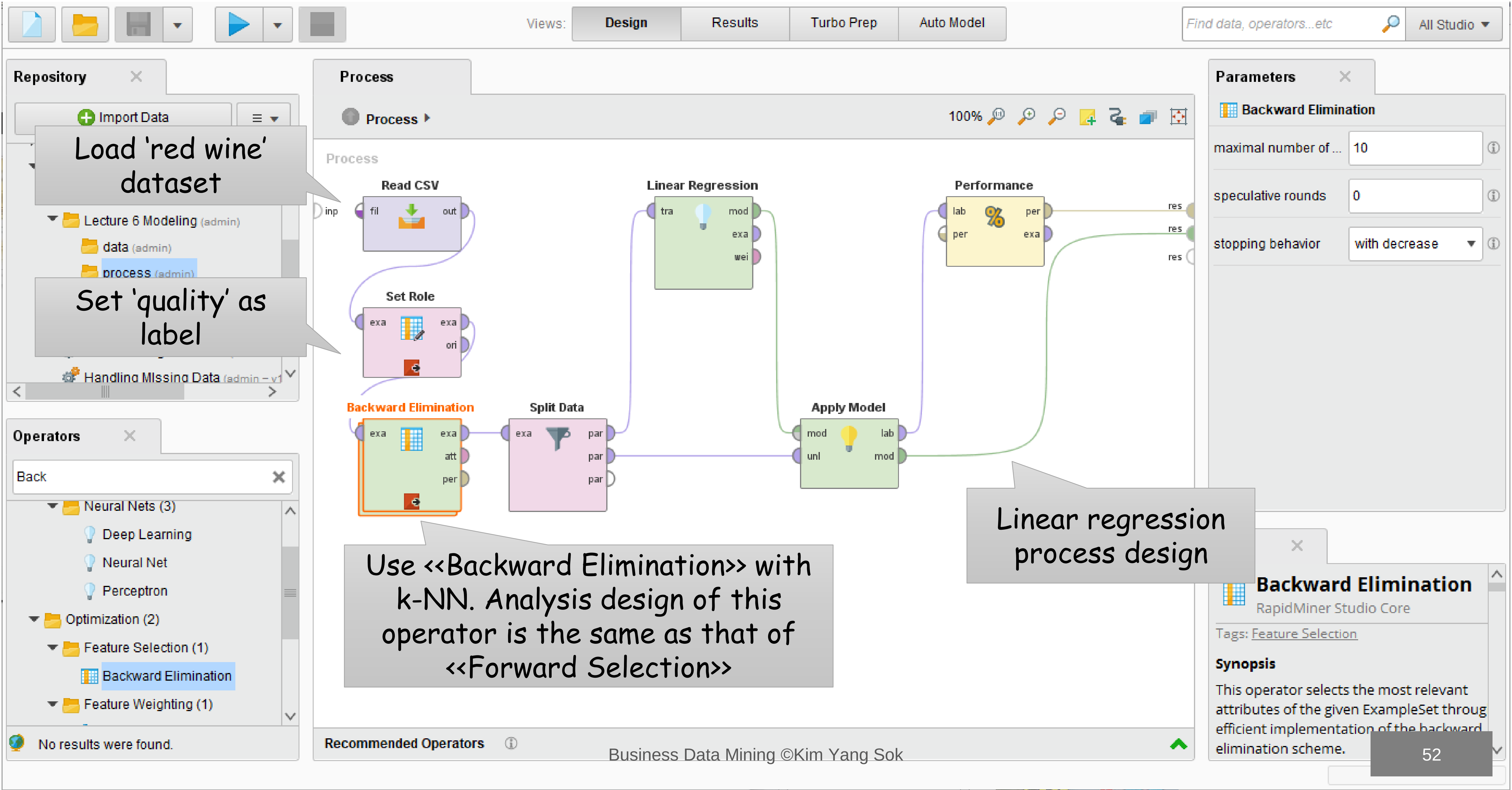
Exercise 7: Linear Regression with Feature Selection using Rapidminer

Task & Process

- **Task**
  - After loading data, build a linear regression model with **backward elimination**.

- **Process**
  - Load “red wine” dataset and Set “quality” as label
  - Select attributes with **backward elimination** approach
  - Perform a linear regression modeling with the split test design
  - Run the analysis process and evaluate analysis results

Perform a linear regression modeling with the split test design



Run the analysis process and evaluate analysis results

Views: Design | Results | Turbo Prep | Auto Model

Result History | ExampleSet (Set Role)

Open in Turbo Prep | Auto Model

Filter (1,599 / 1,599 examples): all

Row No.	quality	fixed acidity	volatile acidity	citric acid	residual sug...	chlorides	free sulfur d...	total sulfur d...	density	pH	sulphates	alcohol
1	5	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400
2	5	7.800	0.880	0	2.600	0.098	25	67	0.997	3.200	0.680	9.800

//Local Repository/Business Data Mining/Lecture 8 Linear Regression/Modeling with Linear Regression Using Backward Elimination – RapidMiner Studio Educational 9.3.001 @ DESKTOP-TPCESUJ

File Edit Process View Connections Settings Extensions Help

Views: Design | Results | Turbo Prep | Auto Model

Result History | ExampleSet (Backward Elimination)

Open in Turbo Prep | Auto Model

Filter (1,599 / 1,599 examples): all

Row No.	quality	fixed acidity	volatile acidity	citric acid	residual sug...	chlorides	total sulfur d...	density	pH	sulphates	alcohol
1	5	7.400	0.700	0	1.900	0.076	34	0.998	3.510	0.560	9.400
2	5	7.800	0.880	0	2.600	0.098	67				9.800
3	5	7.800	0.760	0.040	2.300	0.092	54				9.800
4	6	11.200	0.280	0.560	1.900	0.075	60				9.800
5	5	7.400	0.700	0	1.900	0.076	34				9.400
6	5	7.400	0.660	0	1.800	0.075	40				9.400
7	5	7.900	0.600	0.060	1.600	0.069	59	0.996	3.300	0.460	9.400
8	7	7.300	0.650	0	1.200	0.065	21	0.995	3.390	0.470	10

After applying backward elimination, 10 attributes are selected except 'free sulfur dioxide'.

Run the analysis process and evaluate analysis results

Views: Design | Results | Turbo Prep | Auto Model

Find data, operators...etc | All Studio

Result History | LinearRegression (Linear Regression) | PerformanceVector (Performance)

Data

Description

Annotations

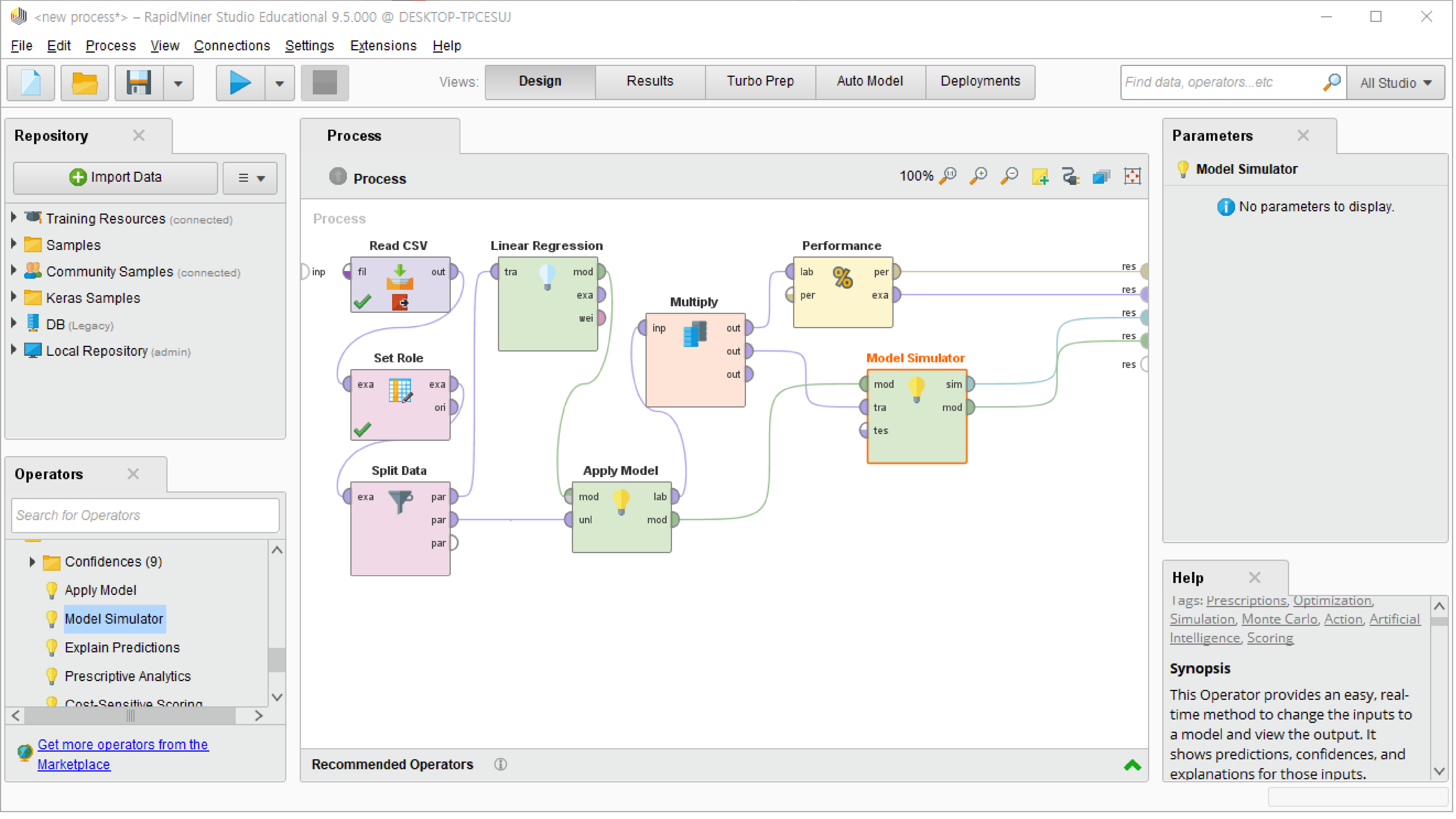
Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
fixed acidity	-0.008	0.017	-0.016	0.950	-0.468	0.640	
volatile acidity	-1.202	0.120	-0.263	0.859	-9.999	0	****
residual sugar	0.016	0.014	0.027	1.000	1.086	0.278	
chlorides	-1.786	0.458	-0.109	0.994	-3.904	0.000	****
total sulfur dioxide	-0.002	0.001	-0.089	0.973	-3.516	0.000	****
pH	-0.482	0.193	-0.089	0.998	-2.491	0.013	**
sulphates	0.879	0.140	0.174	0.949	6.294	0.000	****
alcohol	0.291	0.020	0.386	0.871	14.598	0	****
(Intercept)	4.517	0.755	?	?	5.986	0.000	****

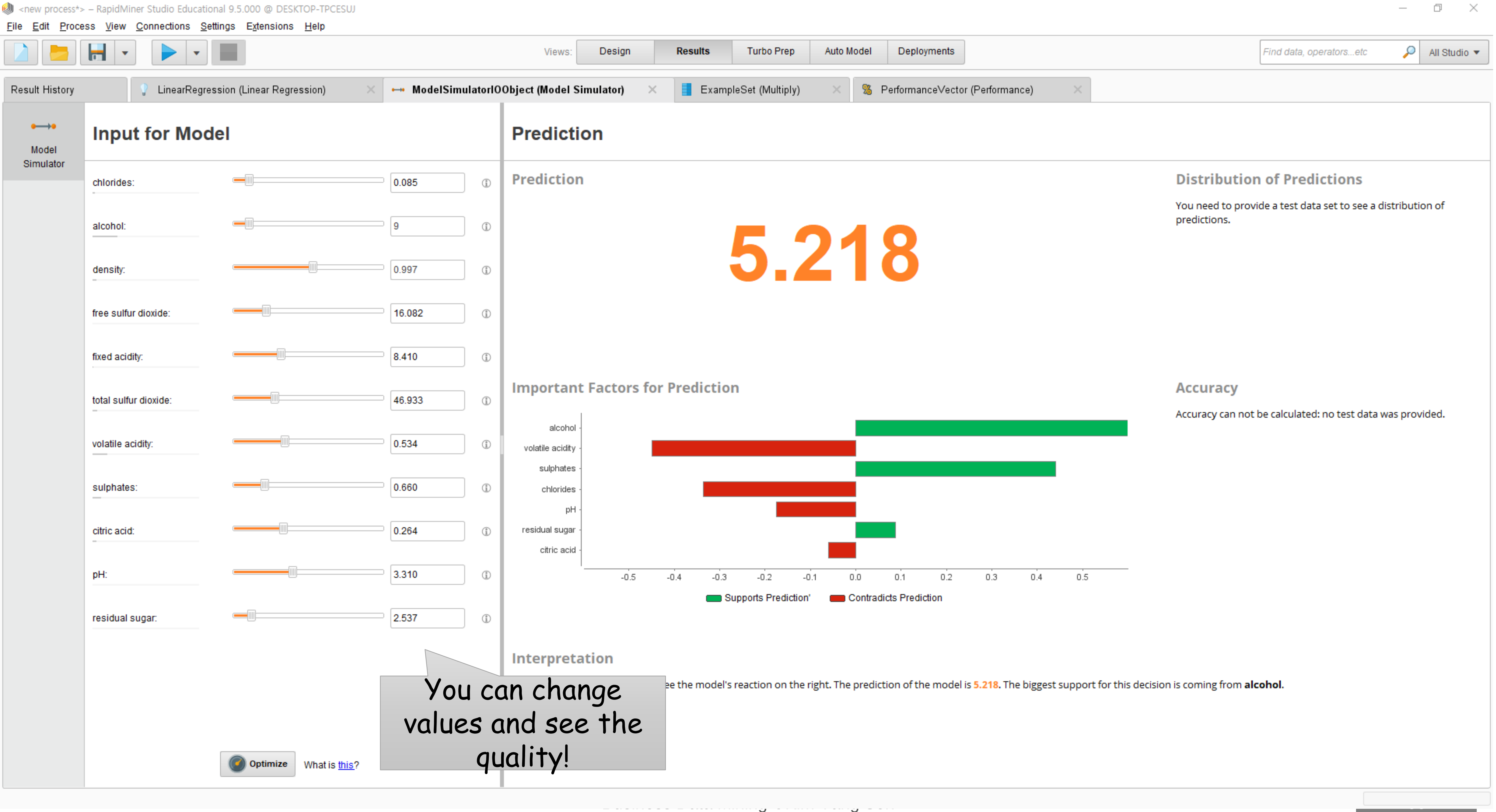
```
PerformanceVector:
root_mean_squared_error: 0.659 +/- 0.000
absolute_error: 0.509 +/- 0.419
relative_error: 8.99% +/- 7.92%
```

Exercise 9:
Model Simulator
with Linear Regression

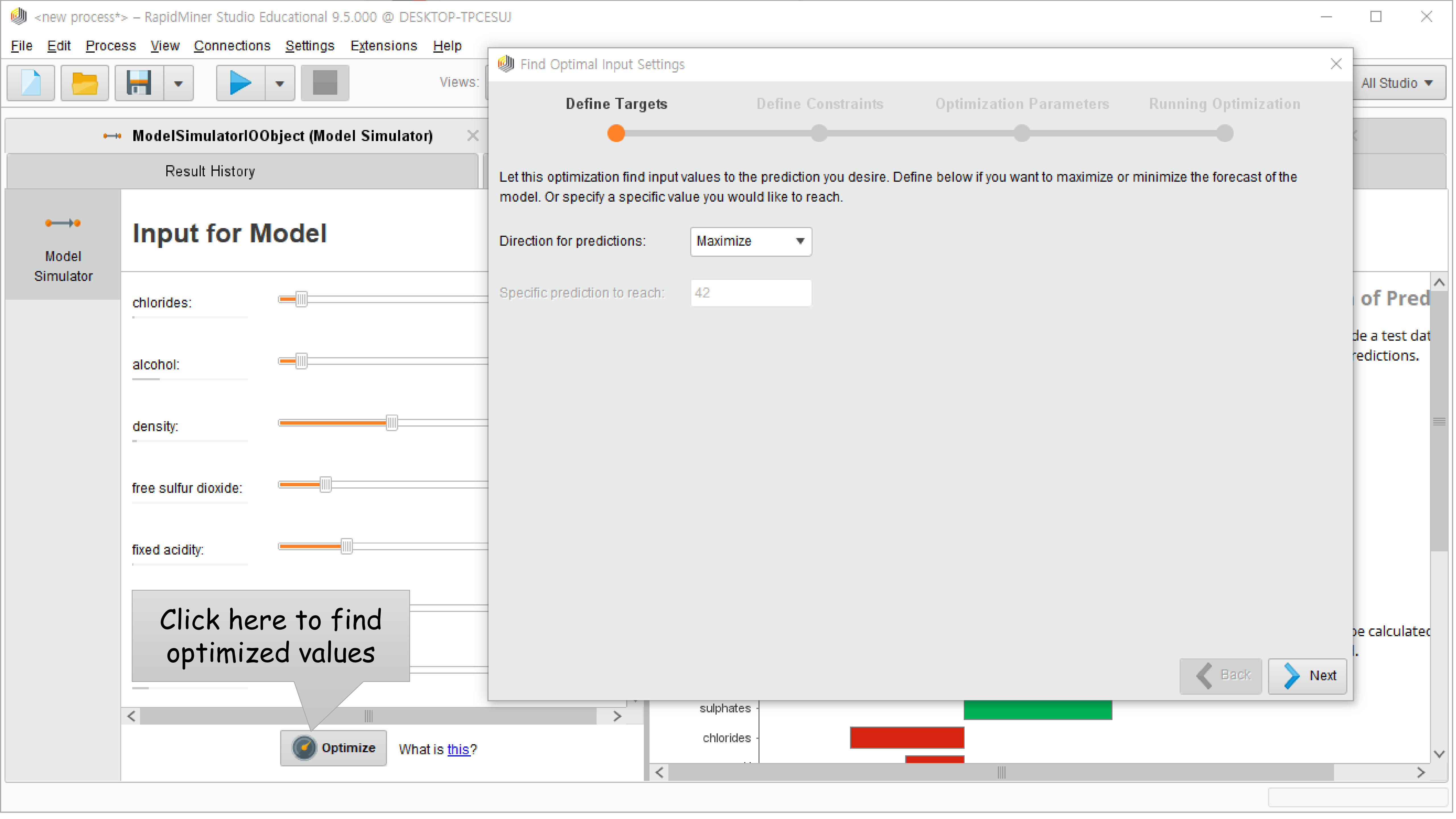
Exercise 9: Model Simulator with Linear Regression

Task & Process

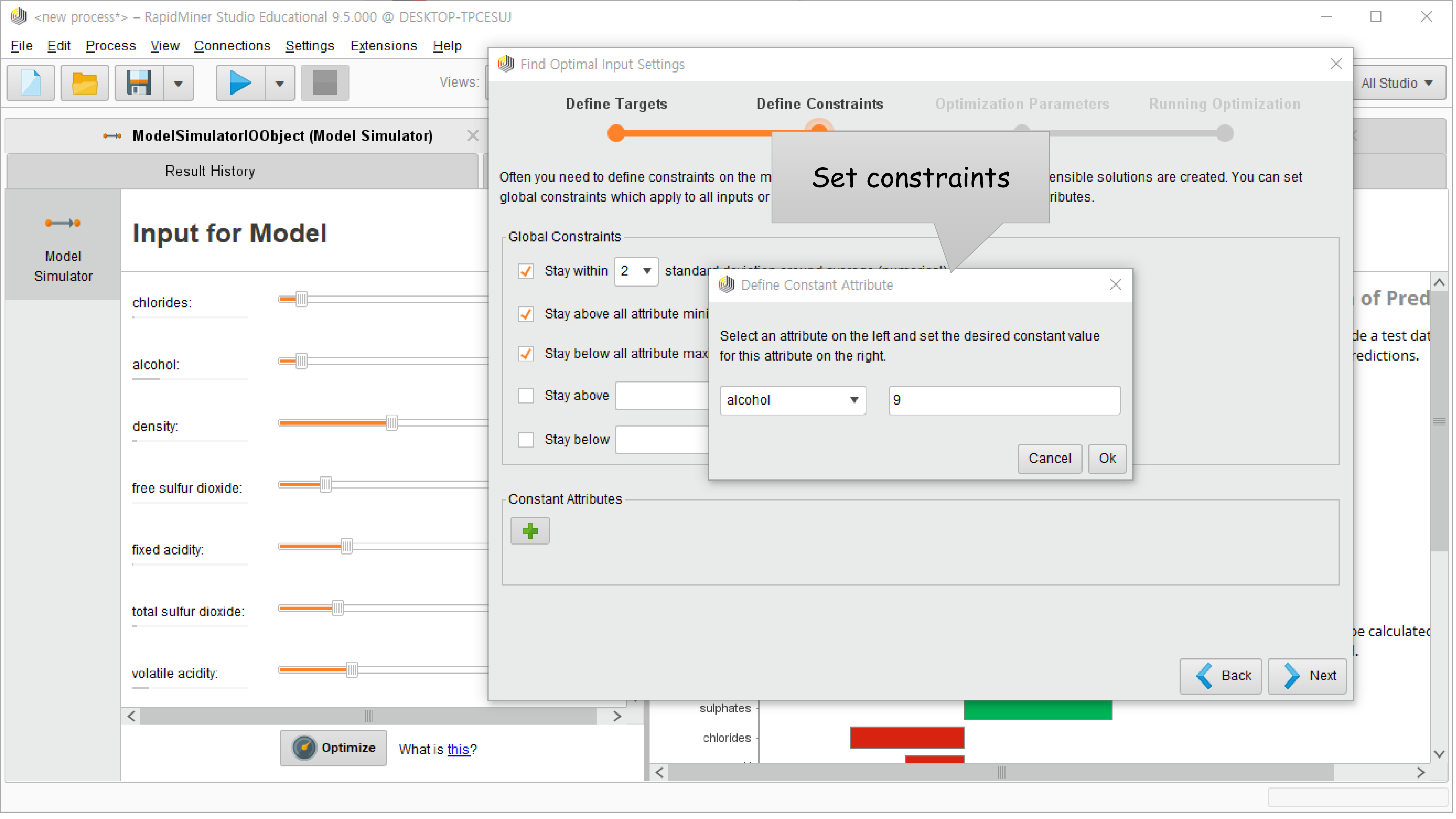

<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Repository
Import Data
Training Resources (connected)
Samples
Community Samples (connected)
Keras Samples
DB (Legacy)
Local Repository (admin)
Process
100%
Read CSV
Linear Regression
Performance
Multiply
Set Role
Model Simulator
Split Data
Apply Model
Operators
Search for Operators
Confidences (9)
Apply Model
Model Simulator
Explain Predictions
Prescriptive Analytics
Cost-Sensitive Scoring
Get more operators from the Marketplace
Recommended Operators
Parameters
Model Simulator
No parameters to display.
Help
Tags: Prescriptions, Optimization, Simulation, Monte Carlo, Action, Artificial Intelligence, Scoring
Synopsis
This Operator provides an easy, real-time method to change the inputs to a model and view the output. It shows predictions, confidences, and explanations for those inputs.

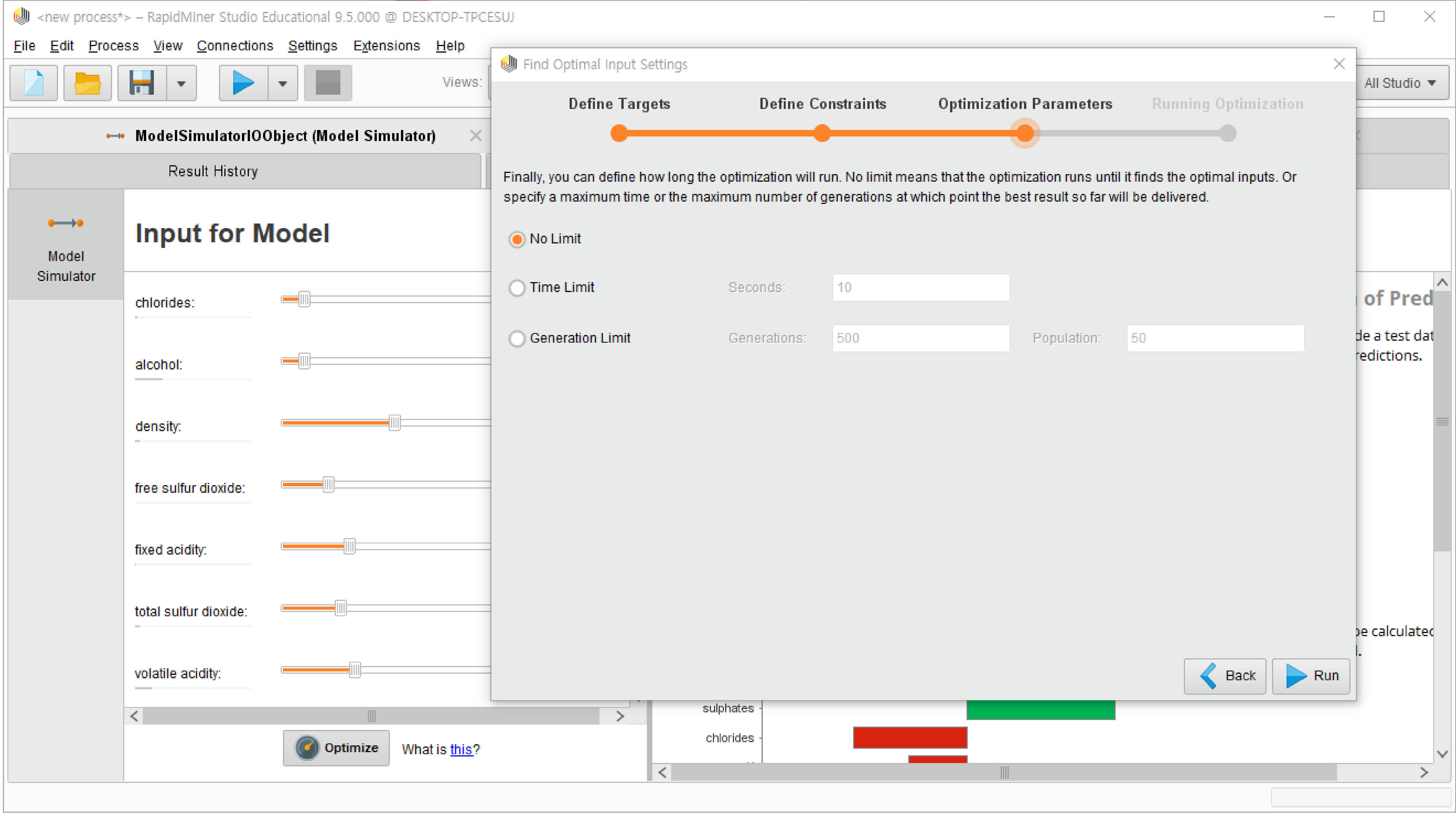


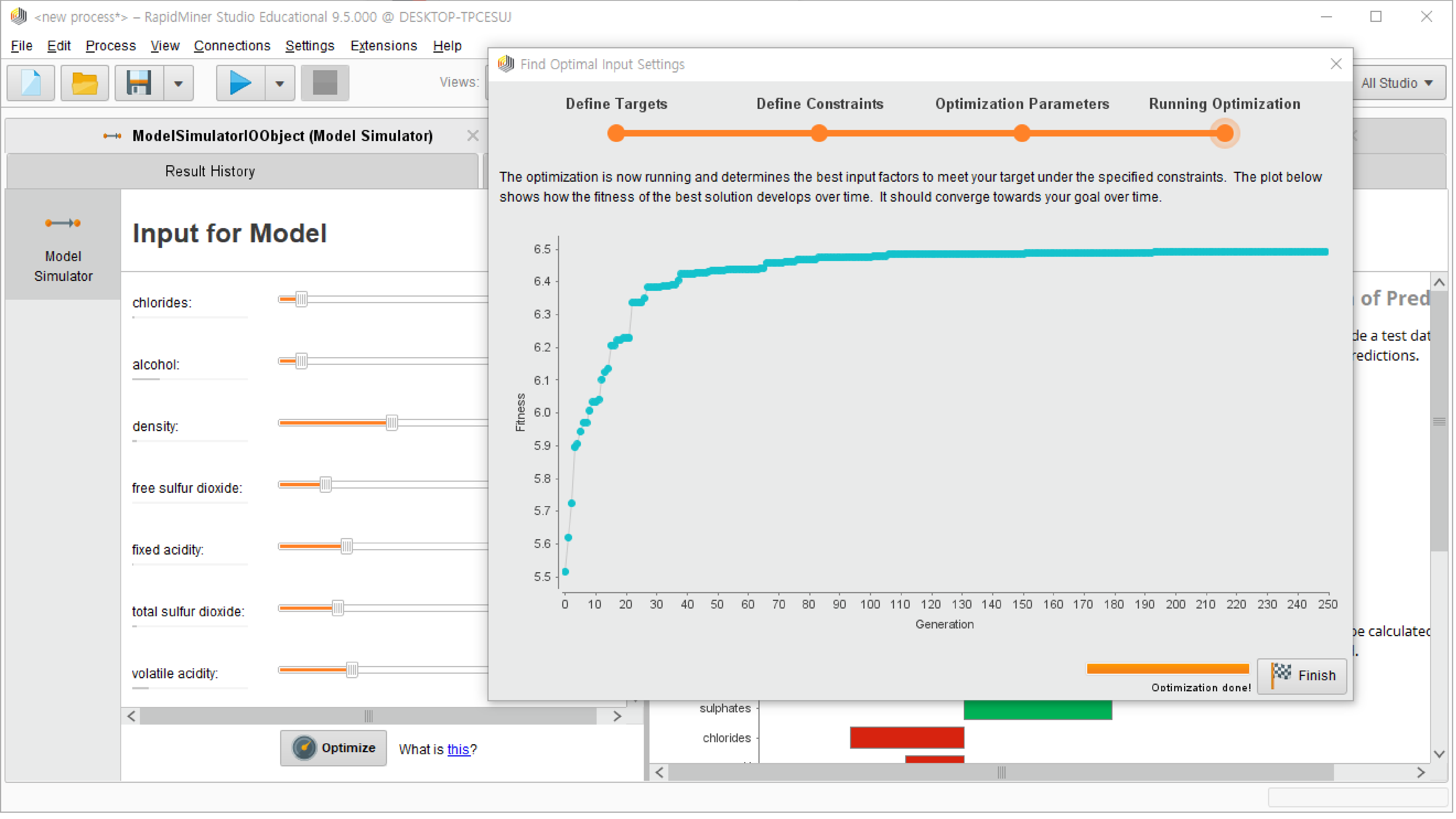
You can change values and see the quality!



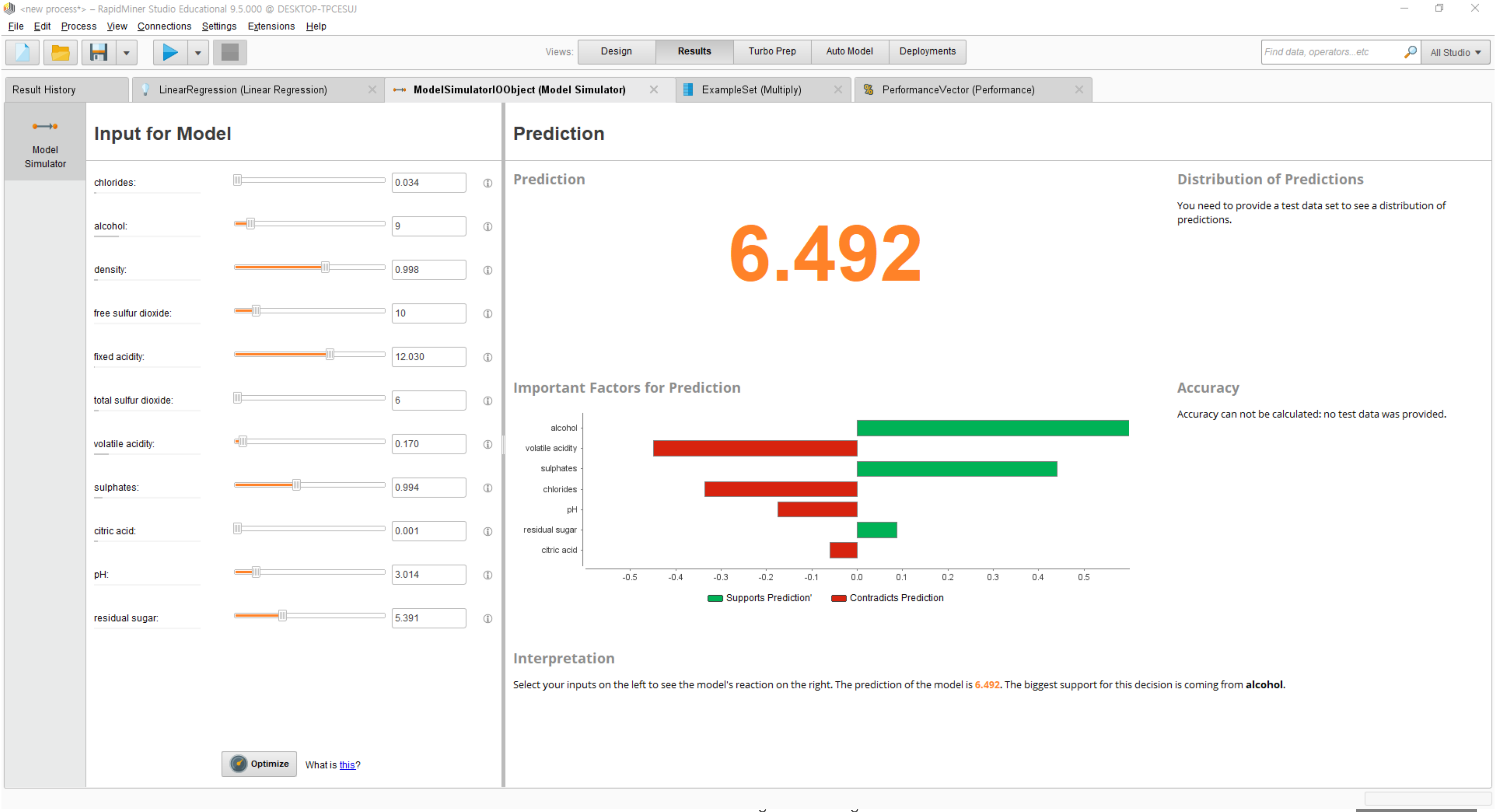
<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File
Edit
Process
View
Connections
Settings
Extensions
Help
Views:
All Studio
ModelSimulatorIOObject (Model Simulator)
Result History
Model Simulator
Input for Model
chlorides:
alcohol:
density:
free sulfur dioxide:
fixed acidity:
Click here to find optimized values
Optimize
What is this?
Find Optimal Input Settings
Define Targets
Define Constraints
Optimization Parameters
Running Optimization
Let this optimization find input values to the prediction you desire. Define below if you want to maximize or minimize the forecast of the model. Or specify a specific value you would like to reach.
Direction for predictions:
Maximize
Specific prediction to reach:
42
Back
Next
sulphates
chlorides


<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File Edit Process View Connections Settings Extensions Help
Views:
All Studio
ModelSimulatorIOObject (Model Simulator)
Result History
Model Simulator
Input for Model
chlorides:
alcohol:
density:
free sulfur dioxide:
fixed acidity:
total sulfur dioxide:
volatile acidity:
Optimize
What is this?
Find Optimal Input Settings
Define Targets
Define Constraints
Optimization Parameters
Running Optimization
Set constraints
Often you need to define constraints on the m
ensible solutions are created. You can set
global constraints which apply to all inputs or
ributes.
Global Constraints
Stay within
2
standar
Stay above all attribute mini
Stay below all attribute max
Stay above
Stay below
Define Constant Attribute
Select an attribute on the left and set the desired constant value for this attribute on the right.
alcohol
9
Cancel
Ok
Constant Attributes
Back
Next
of Pred
de a test dat
redictions.
be calculated
sulphates
chlorides


<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File Edit Process View Connections Settings Extensions Help
Views:
All Studio
ModelSimulatorIOObject (Model Simulator)
Result History
Model Simulator
Input for Model
chlorides:
alcohol:
density:
free sulfur dioxide:
fixed acidity:
total sulfur dioxide:
volatile acidity:
Optimize
What is this?
Find Optimal Input Settings
Define Targets
Define Constraints
Optimization Parameters
Running Optimization
Finally, you can define how long the optimization will run. No limit means that the optimization runs until it finds the optimal inputs. Or specify a maximum time or the maximum number of generations at which point the best result so far will be delivered.
No Limit
Time Limit
Seconds:
10
Generation Limit
Generations:
500
Population:
50
Back
Run
sulphates
chlorides



<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File Edit Process View Connections Settings Extensions Help
Views:
All Studio
ModelSimulatorIOObject (Model Simulator)
Result History
Model Simulator
Input for Model
chlorides:
alcohol:
density:
free sulfur dioxide:
fixed acidity:
total sulfur dioxide:
volatile acidity:
Optimize
What is this?
Find Optimal Input Settings
Define Targets
Define Constraints
Optimization Parameters
Running Optimization
The optimization is now running and determines the best input factors to meet your target under the specified constraints. The plot below shows how the fitness of the best solution develops over time. It should converge towards your goal over time.
Fitness
Generation
Optimization done!
Finish
sulphates
chlorides


<new process*> – RapidMiner Studio Educational 9.5.000 @ DESKTOP-TPCESUJ
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Result History
LinearRegression (Linear Regression)
ModelSimulatorIOObject (Model Simulator)
ExampleSet (Multiply)
PerformanceVector (Performance)
Model Simulator
Input for Model
chlorides: 0.034
alcohol: 9
density: 0.998
free sulfur dioxide: 10
fixed acidity: 12.030
total sulfur dioxide: 6
volatile acidity: 0.170
sulphates: 0.994
citric acid: 0.001
pH: 3.014
residual sugar: 5.391
Optimize
What is this?
Prediction
Prediction
6.492
Distribution of Predictions
You need to provide a test data set to see a distribution of predictions.
Important Factors for Prediction
alcohol
volatile acidity
sulphates
chlorides
pH
residual sugar
citric acid
-0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5
Supports Prediction'
Contradicts Prediction
Accuracy
Accuracy can not be calculated: no test data was provided.
Interpretation
Select your inputs on the left to see the model's reaction on the right. The prediction of the model is 6.492. The biggest support for this decision is coming from alcohol.

Conclusion

Conclusion

- In this lecture, we focused on Linear Regression algorithm.
  - This algorithm derives a linear equation that represents the data
- We also learn also various feature selection algorithms.
- In the next lectures, you will learn optimization in model building with feature extraction algorithms.

QUESTIONS?