

ADDITIONAL EXPERIMENT – 2

CASE STUDY – 4:

TITLE: - Predicting Diabetes in Patients

Problem Statement:

A healthcare provider wants to predict whether a patient is at risk of developing diabetes.

Dataset:

Pima Indians Diabetes Dataset (UCI Machine Learning Repository), containing patient details like glucose levels, blood pressure, BMI, and age.

Approach Using Rapid Miner:

- 1. Data Preprocessing:** Handle missing values, normalize features, and remove outliers.
- 2. Feature Selection:** Identify important variables like glucose concentration and insulin levels.
- 3. Modeling:** Train Decision Trees, Support Vector Machines (SVM), and Neural Networks.
- 4. Evaluation:** Compare models using AUC-ROC, precision, recall, and F1-score.

Outcome:

Achieved 80%+ accuracy in predicting diabetes risk, enabling early intervention.

Step 1: Importing Dataset

In the RapidMiner environment, click on the **Design** option at the top. On the left panel, import the **Pima Indians Diabetes Dataset**. If the dataset is in CSV format, use "**Read CSV**" to load it. Drag the dataset to the environment, connect it to the **output**, and run the process to verify data import.

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Interactive Analysis

Find data, operators, etc. All Studio

Repository

Import Data

- Training Resources (connected)
- Samples
- Local Repository (Local)
- Community Samples (connected)
- DB (Legacy)

Operators

read

- Data Access (27)
 - Files (14)
 - Read (13)
 - Read CSV
 - Read Excel
 - Read URL
 - Read Access
 - Read SPSS
 - Read Stata

We found "Cognite Data Fusion Connectors", "Brancube Connector" and 7 more results in the Marketplace. [Show me!](#)

Process

Process

Read CSV

Parameters

Read CSV

Import Configuration Wizard...

csv file

column separators

☐ trim lines

☐ multiline text

☒ use quotes

quotes character

escape character

[Hide advanced parameters](#)

[Change compatibility \(11.0.0.00\)](#)

Help

Read CSV

AI Studio Core

Tags: Load, Import, Read, Data, Files, Text, Commas, Spreadsheet, Excel, Datasets, Tsv

Synopsis

This Operator reads an ExampleSet from the specified CSV file.

[Jump to Tutorial Process](#)

Description

CSV is an abbreviation for Comma-Separated Values. The CSV files

Leverage the Wisdom of Crowds to get operator recommendations based on your process design

☒ Activate Wisdom of Crowds

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Interactive Analysis

Find data, operators, etc. All Studio

Result History

ExampleSet (Read CSV)

Open in Turbo Prep Auto Model Interactive Analysis

Filter (768 / 768 examples): all

Row No.	Pregnancies	Glucose	BloodPress...	SkinThickne...	Insulin	BMI	DiabetesPe...	Age	Outcome
1	6	148	72	35	0	33.600	0.627	50	1
2	1	85	66	29	0	26.600	0.351	31	0
3	8	183	64	0	0	23.300	0.672	32	1
4	1	89	66	23	94	28.100	0.167	21	0
5	0	137	40	35	168	43.100	2.288	33	1
6	5	116	74	0	0	25.600	0.201	30	0
7	3	78	50	32	88	31	0.248	26	1
8	10	115	0	0	0	35.300	0.134	29	0
9	2	197	70	45	543	30.500	0.158	53	1
10	8	125	96	0	0	0	0.232	54	1
11	4	110	92	0	0	37.600	0.191	30	0
12	10	168	74	0	0	38	0.537	34	1
13	10	139	80	0	0	27.100	1.441	57	0
14	1	189	60	23	846	30.100	0.398	59	1
15	5	166	72	19	175	25.800	0.587	51	1
16	7	100	0	0	0	30	0.484	32	1
17	0	118	84	47	230	45.800	0.551	31	1
18	7	107	74	0	0	29.600	0.254	31	1
19	1	103	30	38	83	43.300	0.183	33	0
20	1	115	70	30	96	34.600	0.529	32	1
21	3	126	88	41	235	39.300	0.704	27	0

ExampleSet (768 examples, 0 special attributes, 9 regular attributes)

Repository

Import Data

- Training Resources (connected)
- Samples
- Local Repository (Local)
- Community Samples (connected)
- DB (Legacy)

Step 2: Converting Numerical to Binominal

Since the **Outcome** attribute has values **0** and **1**, it needs to be converted into a binominal type:

1. Search for "**Numerical to Binominal**" in the operators search bar.
2. Drag the operator to the environment.
3. Connect the dataset output to the **Numerical to Binominal** input.
4. In the **Parameters** panel, select **Outcome** as the attribute to convert.
5. Connect the output to **Results** and run to verify the conversion.

The screenshot displays the Al Studio interface with the following components:

- Repository:** Shows a tree view with 'Training Resources (connected)', 'Samples', 'Local Repository (Local)', 'Community Samples (connected)', and 'DB (Legacy)'.
- Operators:** A search bar contains 'nu'. The results list various operators under 'Blending (14)', 'Attributes (13)', and 'Types (12)'. The 'Numerical to Binominal' operator is highlighted.
- Process:** A workflow diagram showing a 'Read CSV' operator connected to a 'Numerical to Binominal' operator. The output of the 'Numerical to Binominal' operator is connected to a 'Results' output.
- Parameters:** A panel for the 'Numerical to Binominal' operator. It includes:
 - attribute filter type:** single
 - attribute:** Outcome
 - invert selection:** unchecked
 - include special attributes:** unchecked
 - min:** 0.0
 - max:** 0.0
- Help:** A panel for the 'Numerical to Binominal' operator. It includes:
 - Tags:** Binary, Binarizer, Dual, Categorical, Continuous, Types
 - Synopsis:** This operator changes the type of the selected numeric attributes to a binominal type. It also maps all values of these attributes to corresponding binominal values.
 - Description:** (Empty)

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Interactive Analysis

Result History ExampleSet (Numerical to Binominal) X

Open in Turbo Prep Auto Model Interactive Analysis Filter (768 / 768 examples): all

Row No.	Outcome	Pregnancies	Glucose	BloodPress...	SkinThickne...	Insulin	BMI	DiabetesPe...	Age
1	true	6	148	72	35	0	33.600	0.627	50
2	false	1	85	66	29	0	26.600	0.351	31
3	true	8	183	64	0	0	23.300	0.672	32
4	false	1	89	66	23	94	28.100	0.167	21
5	true	0	137	40	35	168	43.100	2.288	33
6	false	5	116	74	0	0	25.600	0.201	30
7	true	3	78	50	32	88	31	0.248	26
8	false	10	115	0	0	0	35.300	0.134	29
9	true	2	197	70	45	543	30.500	0.158	53
10	true	8	125	96	0	0	0	0.232	54
11	false	4	110	92	0	0	37.600	0.191	30
12	true	10	168	74	0	0	38	0.537	34
13	false	10	139	80	0	0	27.100	1.441	57
14	true	1	189	60	23	846	30.100	0.398	59
15	true	5	166	72	19	175	25.800	0.587	51
16	true	7	100	0	0	0	30	0.484	32
17	true	0	118	84	47	230	45.800	0.551	31
18	true	7	107	74	0	0	29.600	0.254	31
19	false	1	103	30	38	83	43.300	0.183	33
20	true	1	115	70	30	96	34.600	0.529	32
21	false	3	126	88	41	235	39.300	0.704	27

ExampleSet (768 examples, 0 special attributes, 9 regular attributes)

Step 3: Data Preprocessing-

After converting numerical to binominal, perform the following preprocessing steps:

Replacing Missing Values:

1. Search for "**Replace Missing Values**", drag it to the environment.
2. Connect the **Numerical to Binominal** output to **Replace Missing Values**.
3. In the **Parameters panel**, set missing values to be replaced with the **median**.

Normalize Data:

1. Search for "**Normalize**", drag it to the environment.
2. Connect the **Replace Missing Values** output to **Normalize**.
3. Choose **Min-Max Scaling** or **Z-Score Normalization** to bring numeric values into a standard range.

4. Connect the output to **Results** and observe the cleaned dataset.

The screenshot shows the AI Studio interface with the Design view selected. The workflow consists of four operators: Read CSV, Numerical to Binomial, Replace Missing Values, and Normalize. The Normalize operator is selected, and its parameters are shown on the right. The Parameters panel for the Normalize operator shows 'attribute filter type' set to 'all', 'invert selection' unchecked, 'include special attributes' unchecked, and 'method' set to 'Z-transformation'. The Help panel for the Normalize operator is also visible, providing a synopsis and description of the operator.

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Interactive Analysis

Result History ExampleSet (Normalize)

Open in Turbo Prep Auto Model Interactive Analysis

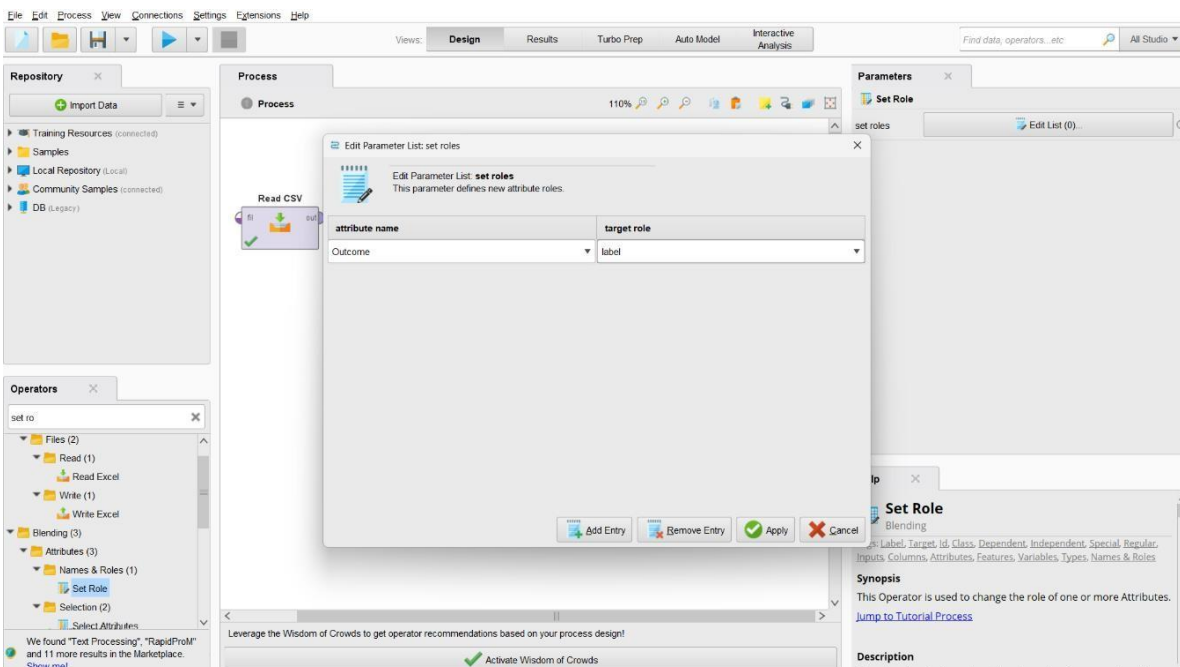
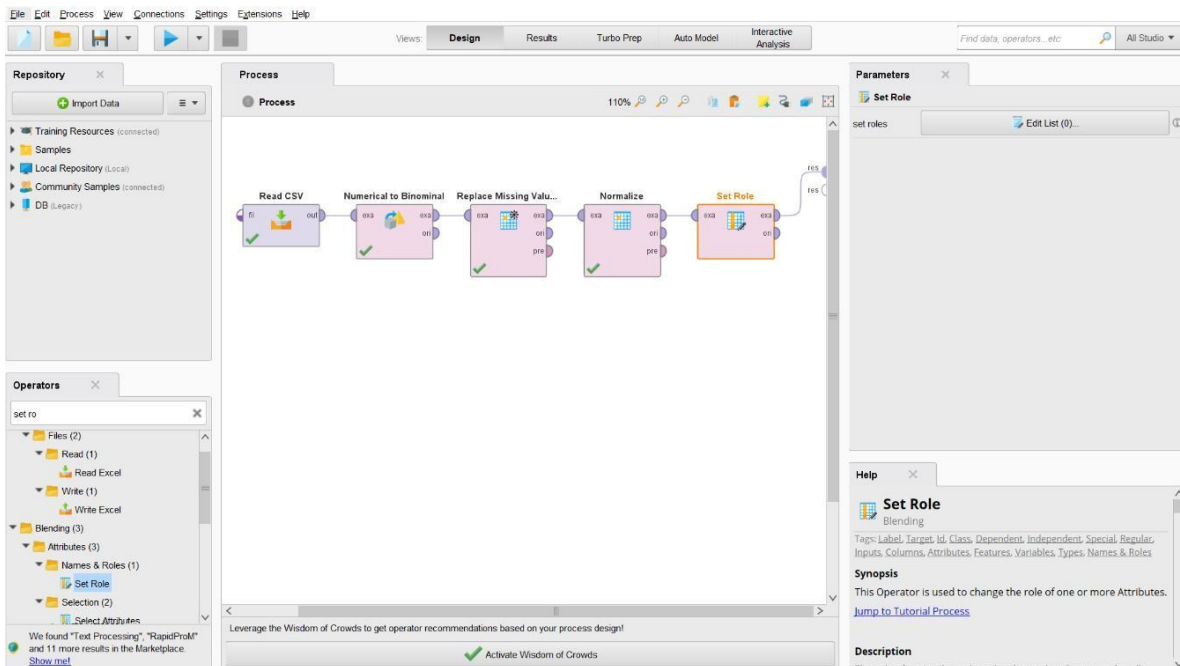
Filter (768 / 768 examples): all

Row No.	Pregnancies	Glucose	BloodPress...	SkinThickne...	Insulin	BMI	DiabetesPe...	Age	Outcome
1	0.640	0.848	0.150	0.907	-0.692	0.204	0.468	1.425	true
2	-0.844	-1.123	-0.160	0.531	-0.692	-0.684	-0.365	-0.191	false
3	1.233	1.942	-0.264	-1.287	-0.692	-1.103	0.604	-0.106	true
4	-0.844	-0.998	-0.160	0.154	0.123	-0.494	-0.920	-1.041	false
5	-1.141	0.504	-1.504	0.907	0.765	1.409	5.481	-0.020	true
6	0.343	-0.153	0.253	-1.287	-0.692	-0.811	-0.818	-0.276	false
7	-0.251	-1.342	-0.987	0.719	0.071	-0.126	-0.676	-0.616	true
8	1.827	-0.184	-3.570	-1.287	-0.692	0.420	-1.020	-0.361	false
9	-0.548	2.380	0.046	1.534	4.019	-0.189	-0.947	1.680	true
10	1.233	0.128	1.389	-1.287	-0.692	-4.058	-0.724	1.765	true
11	0.046	-0.341	1.183	-1.287	-0.692	0.711	-0.848	-0.276	false
12	1.827	1.473	0.253	-1.287	-0.692	0.762	0.197	0.065	true
13	1.827	0.566	0.563	-1.287	-0.692	-0.621	2.925	2.020	false
14	-0.844	2.130	-0.470	0.154	6.649	-0.240	-0.223	2.190	true
15	0.343	1.411	0.150	-0.096	0.826	-0.785	0.347	1.510	true
16	0.936	-0.654	-3.570	-1.287	-0.692	-0.253	0.037	-0.106	true
17	-1.141	-0.091	0.770	1.659	1.303	1.751	0.239	-0.191	true
18	0.936	-0.435	0.253	-1.287	-0.692	-0.303	-0.658	-0.191	true
19	-0.844	-0.560	-2.020	1.095	0.028	1.434	-0.872	-0.020	false
20	-0.844	-0.184	0.046	0.593	0.141	0.331	0.172	-0.106	true
21	-0.251	0.160	0.976	1.283	1.347	0.927	0.701	-0.531	false

ExampleSet (768 examples, 0 special attributes, 9 regular attributes)

Step 4: Set Role

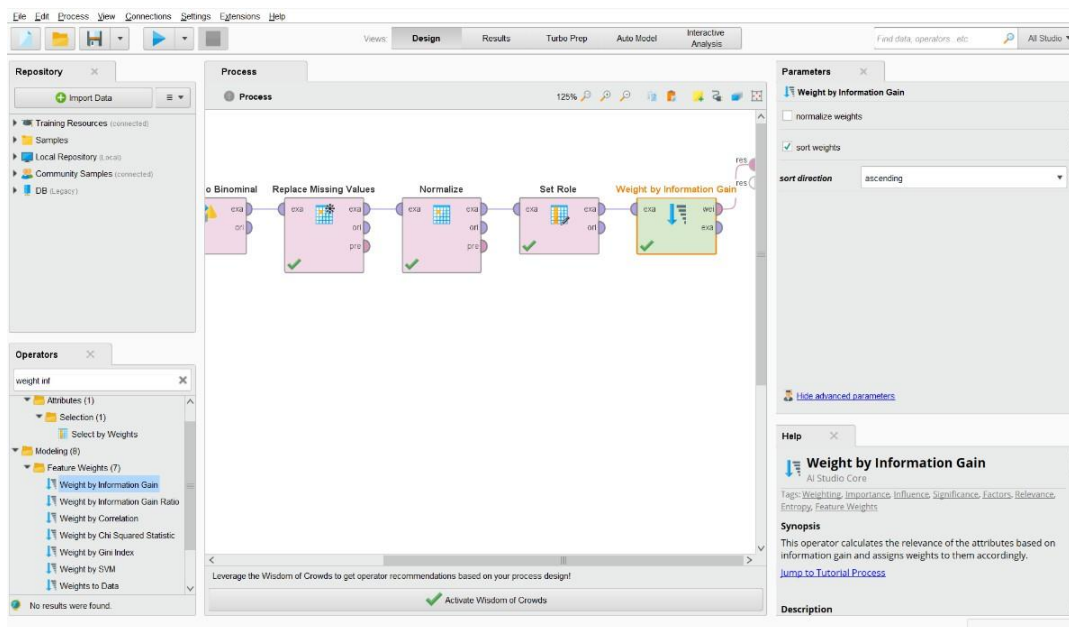
1. Search for **"Set Role"** in the operators panel and drag it to the environment.
2. Connect the **preprocessed data output** to **Set Role**.
3. In the **Parameters** panel, click **"Edit List"**, set **attribute name** to **"Outcome"**, and assign **target role** as **"Label"**.
4. Click **Apply** and connect to the next step.



Step 5: Feature Selection

To improve model performance:

1. Search for "**Weight by Information Gain**", drag it to the environment, and connect it to **Set Role** output.
2. Run the process to check the most important features like **Glucose, BMI, Insulin, and Age**.
3. Use "**Select Attributes**" to keep only the top-ranked features.



The screenshot shows the 'AttributeWeights (Weight by Information Gain)' result in the Al Studio interface. The table displays the following data:

attribute	weight
BloodPre...	0.014
SkinThick...	0.017
Diabetes...	0.021
Insulin	0.027
Pregnanc...	0.039
Age	0.072
BMI	0.075
Glucose	0.131

Step 6: Selecting Attributes

1. Search for "Select Attributes", drag it to the environment.
2. Connect the **Weight by Information Gain** output to **Select Attributes**.
3. In the **Parameters** panel, manually select the **top-ranked features** based on their weights.
4. Connect the **Select Attributes** output to the next step.

The screenshot shows the Orange3 software interface in the 'Design' view. The workflow consists of five operators: 'Using Values', 'Normalize', 'Set Role', 'Weight by Information...', and 'Select Attributes'. The 'Select Attributes' operator is highlighted in orange. The 'Parameters' panel on the right shows settings for 'Select Attributes': type is 'include attributes', attribute filter type is 'all attributes', and 'also apply to special attributes (id, label...)' is unchecked. The 'Operators' panel on the left shows a search for 'select' with 'Select Attributes' selected. The 'Help' panel on the right provides information about the 'Select Attributes' operator.

The screenshot shows the Orange3 software interface in the 'Results' view. The 'Result History' panel displays a table of data for the 'ExampleSet (Select Attributes)' operator. The table has 21 rows and 10 columns: Row No., Outcome, Pregnancies, Glucose, BloodPress..., SkinThicke..., Insulin, BMI, DiabetesPe..., and Age. The data is filtered to show 768 examples.

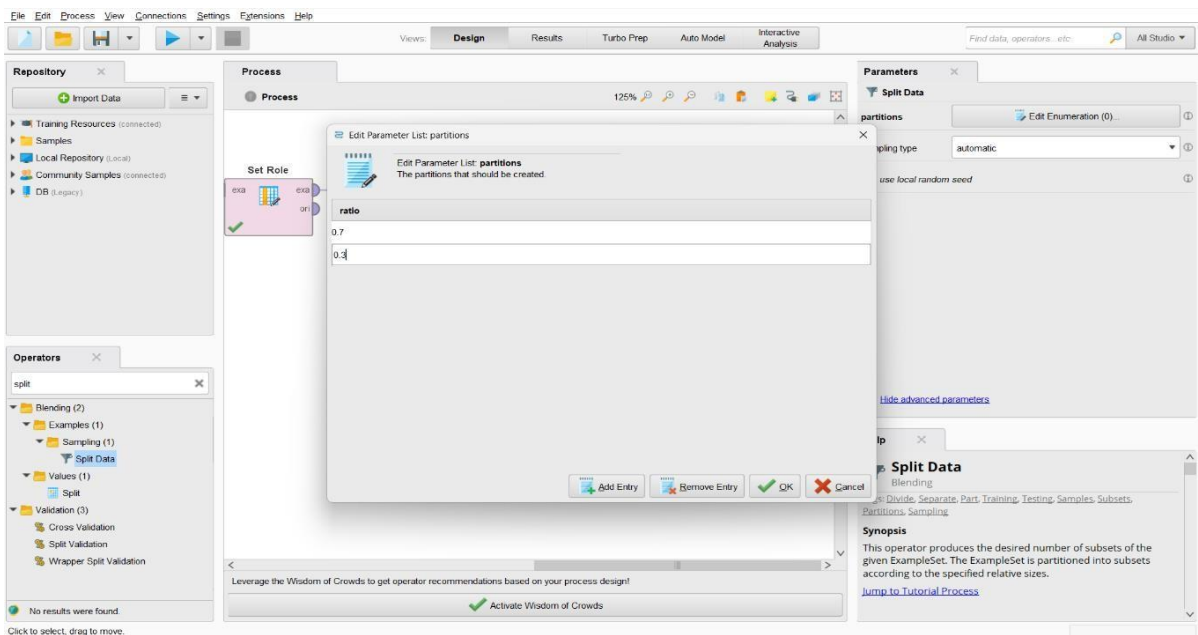
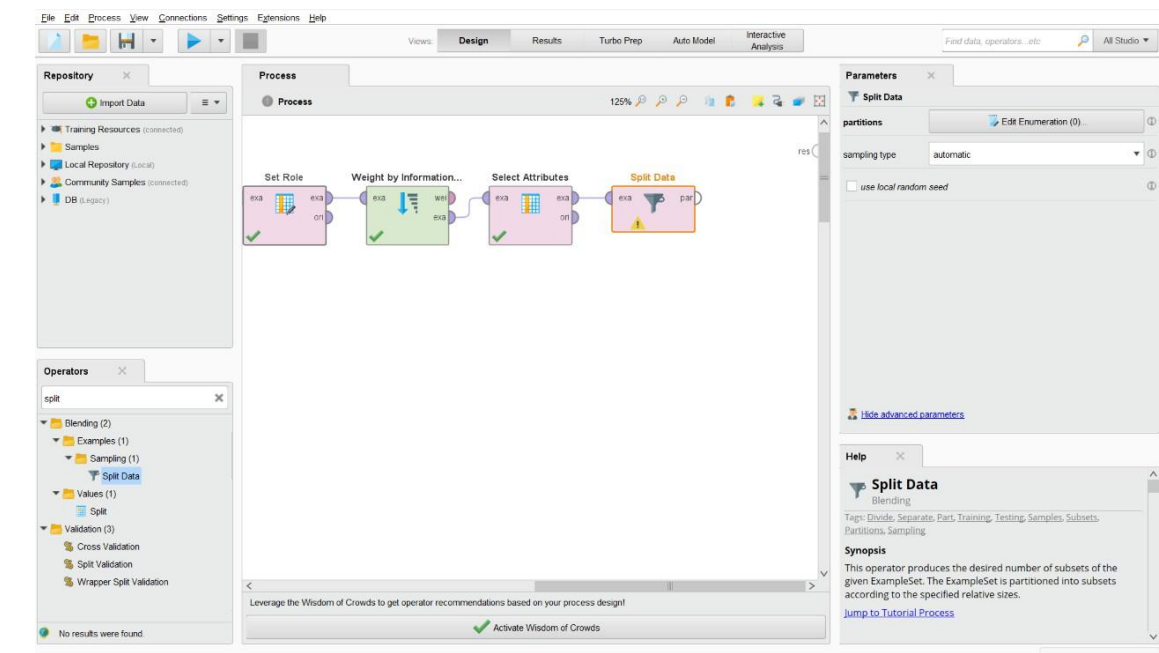
Row No.	Outcome	Pregnancies	Glucose	BloodPress...	SkinThicke...	Insulin	BMI	DiabetesPe...	Age
1	true	0.640	0.848	0.150	0.907	-0.692	0.204	0.468	1.425
2	false	-0.844	-1.123	-0.160	0.531	-0.692	-0.684	-0.365	-0.191
3	true	1.233	1.942	-0.264	-1.287	-0.692	-1.103	0.604	-0.100
4	false	-0.844	-0.998	-0.160	0.154	0.123	-0.494	-0.920	-1.041
5	true	-1.141	0.504	-1.504	0.907	0.765	1.409	5.481	-0.020
6	false	0.343	-0.153	0.253	-1.287	-0.692	-0.811	-0.818	-0.276
7	true	-0.251	-1.342	-0.987	0.719	0.071	-0.126	-0.676	-0.616
8	false	1.827	-0.184	-3.570	-1.287	-0.692	0.420	-1.020	-0.361
9	true	-0.548	2.380	0.046	1.534	4.019	-0.189	-0.947	1.680
10	true	1.233	0.128	1.389	-1.287	-0.692	-4.058	-0.724	1.765
11	false	0.046	-0.341	1.183	-1.287	-0.692	0.711	-0.848	-0.276
12	true	1.827	1.473	0.253	-1.287	-0.692	0.762	0.197	0.065
13	false	1.827	0.568	0.563	-1.287	-0.692	-0.621	2.925	2.020
14	true	-0.844	2.130	-0.470	0.154	6.649	-0.240	-0.223	2.190
15	true	0.343	1.411	0.150	-0.096	0.826	-0.785	0.347	1.510
16	true	0.936	-0.654	-3.570	-1.287	-0.692	-0.253	0.037	-0.106
17	true	-1.141	-0.091	0.770	1.658	1.303	1.751	0.239	-0.191
18	true	0.936	-0.435	0.253	-1.287	-0.692	-0.303	-0.658	-0.191
19	false	-0.844	-0.560	-2.020	1.095	0.028	1.434	-0.872	-0.020
20	true	-0.844	-0.184	0.046	0.593	0.141	0.331	0.172	-0.106
21	false	-0.251	0.160	0.976	1.283	1.347	0.927	0.701	-0.531

ExampleSet (768 examples, 1 special attribute, 8 regular attributes)

Step 7: Splitting Data

To train and test the model:

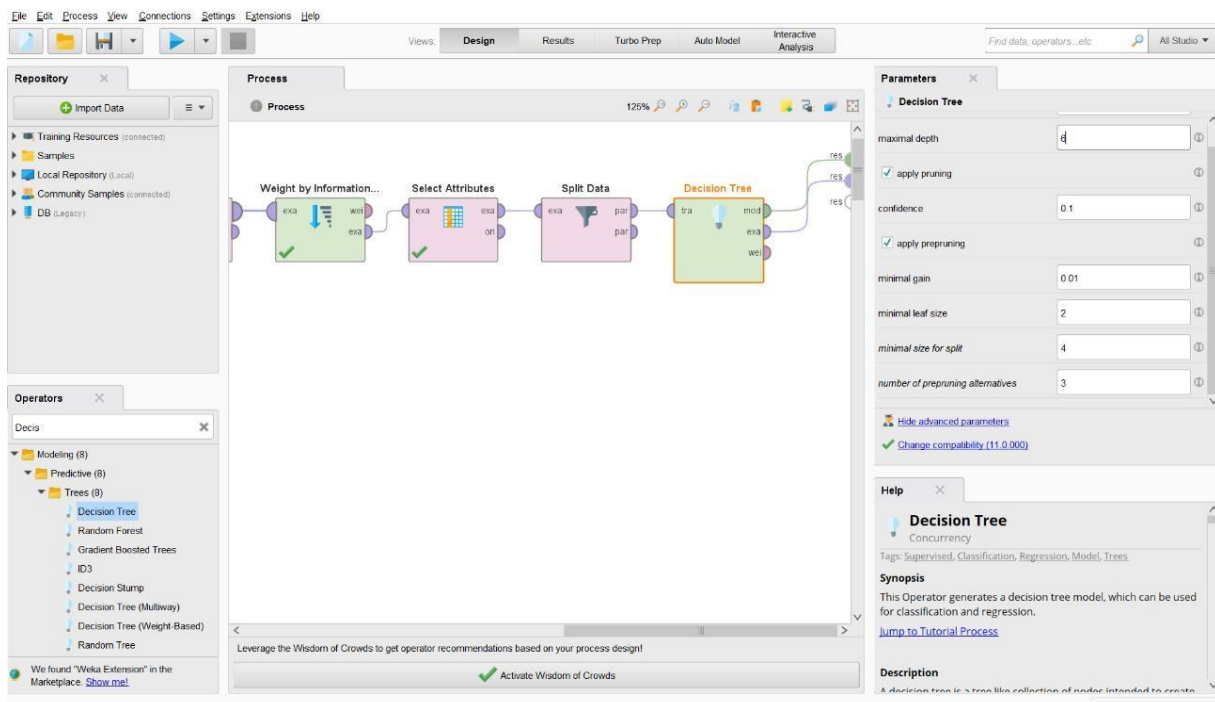
1. Search for **"Split Data"**, drag it to the environment.
2. Connect the **Select Attributes** output to **Split Data**.
3. In the **Parameters** panel, set **training data ratio** to **70% (0.7)** and **testing data ratio** to **30% (0.3)**.
4. The first output will be used for training, and the second output for testing.



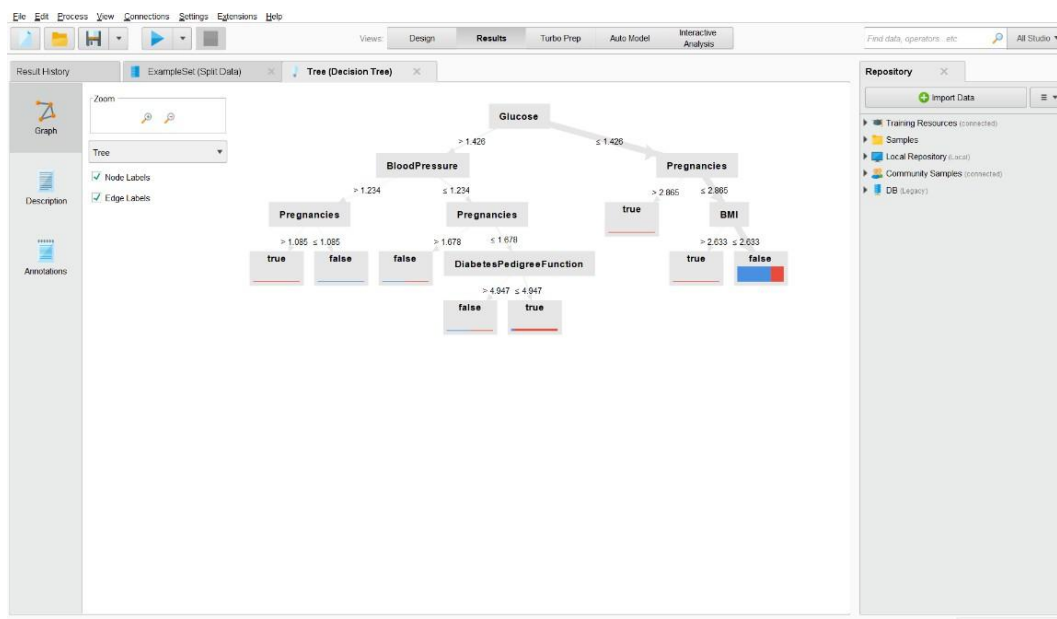
Step 8: Model Training

1. From the operators search bar, add models like **"Decision Tree"**, **"SVM"**, and **"Neural Network"**.
2. Connect the **training data output from Split Data** to the model input.
3. Connect the model output to **Apply Model**.

A. Apply Decision Tree Model



Observe the **Decision Tree** in the statistics we can see the **Decision Tree**.



B. Apply SVM Model:

The screenshot displays the AI Studio interface with the 'Design' view selected. The process flow includes the following operators: Set Role, Weight by Information..., Select Attributes, Split Data, and Decision Tree. The 'Parameters' panel on the right shows the configuration for the 'SVM (Support Vector Machine)' operator:

- kernel type: dot
- kernel cache: 200
- C: 0.0
- convergence epsilon: 0.001
- max iterations: 100000
- scale: ☒
- L pos: 1.0
- L neg: 1.0
- epsilon: 0.0

The 'Help' panel on the right provides information about the 'Support Vector Machine' operator, including its tags (Supervised, Classification, Regression, Model, SVM, Margin, Support Vector Machines) and a synopsis: 'This operator is an SVM (Support Vector Machine) Learner. It is based on the internal Java implementation of the *mySVM* by Stefan Rueping.'

Observe the SVM Results

The screenshot displays the AI Studio interface with the 'Results' view selected. The 'Result History' panel shows the 'Kernel Model (SVM)' operator. The 'Description' tab is active, showing the following results:

Kernel Model

Total number of Support Vectors: 538
Bias (offset): -0.745

Weight Table

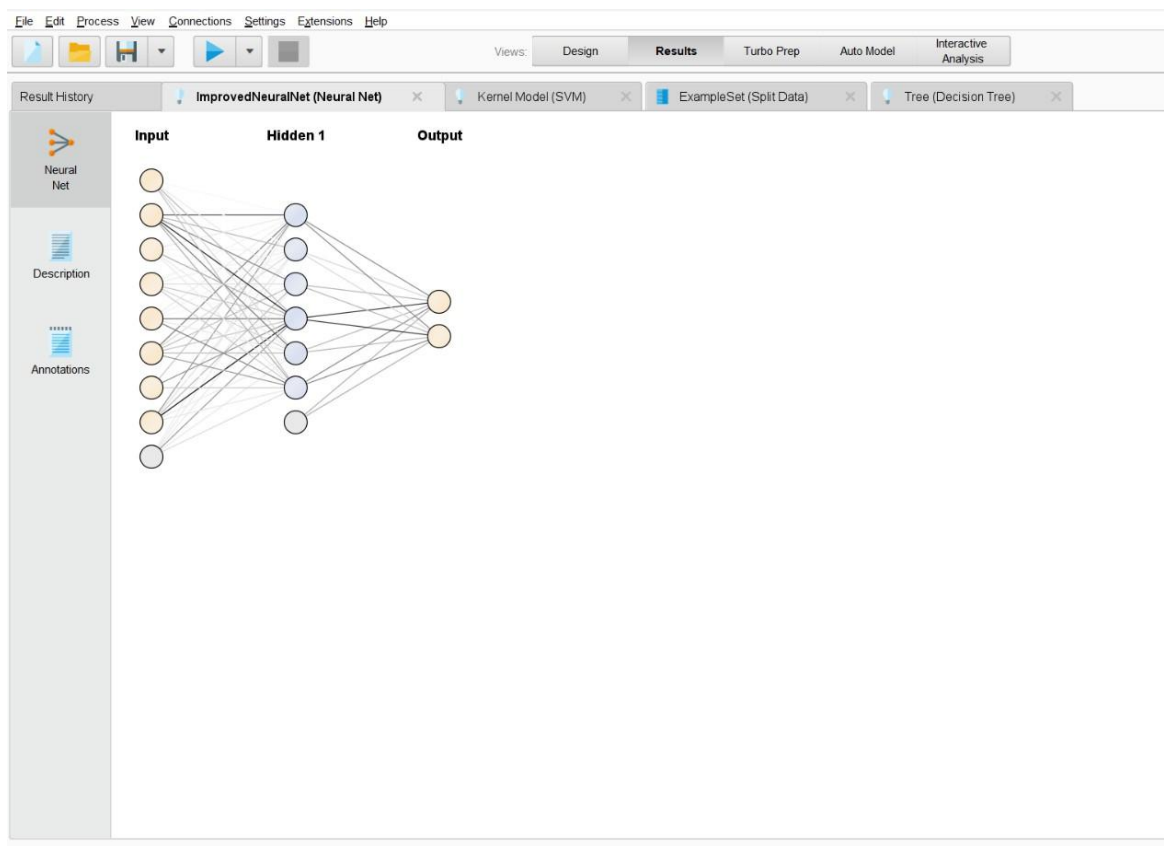
Feature	Weight
w[Pregnancies]	0.374
w[Glucose]	0.797
w[BloodPressure]	-0.264
w[SkinThickness]	0.025
w[Insulin]	-0.016
w[BMI]	0.571
w[DiabetesPedigreeFunction]	0.259
w[Age]	0.032

The 'Support Vector Table' and 'Support Vector Visualization' tabs are also visible in the sidebar.

C. Observe the Neural Network

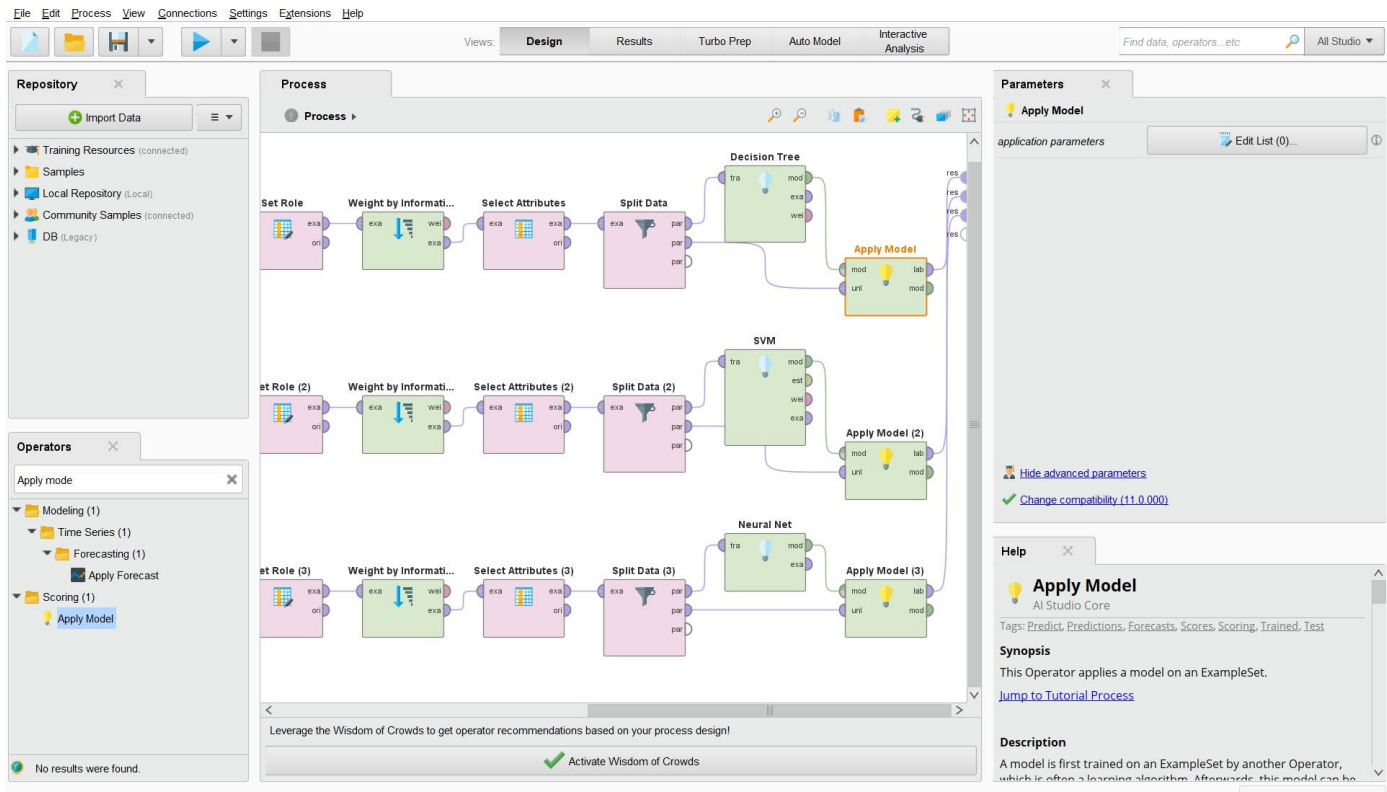
The screenshot displays the AI Studio interface in the Design view. The main workspace shows a process flow with three parallel paths, each starting with a 'Normalize' operator, followed by 'Set Role', 'Weight by Information Gain', 'Select Attributes', 'Split Data', and ending with either a 'Decision Tree' or 'SVM' model. A 'Neural Net' operator is also visible in the process flow. The left sidebar contains the 'Repository' and 'Operators' panels. The 'Operators' panel shows a search for 'Neural' and lists 'Deep Learning', 'Predictive', and 'Neural Nets' with sub-operators like 'Deep Learning', 'Neural Net', 'AutoMLP', and 'Perceptron'. The right sidebar shows the 'Parameters' panel for the 'Neural Net' operator, with settings for hidden layers, training cycles (200), learning rate (0.01), momentum (0.9), decay, shuffle, normalize, error epsilon (1.0E-4), and use local random seed. A 'Help' panel at the bottom right provides a synopsis of the Neural Net operator.

Observe the Neural Network



Step 9: Apply Model

1. Search for "**Apply Model**", drag it to the environment.
2. Connect the **trained model output** to **Apply Model**.
3. Also, connect the **testing data output from Split Data** to **Apply Model**.
4. Run the process and verify predictions.



Step 10: Measuring Performance

1. Search for "**Performance (Binominal Classification)**", drag it to the environment.
2. Connect **Apply Model** output to **Performance**.
3. In the **Parameters** panel, select **Accuracy, Precision, Recall, AUC-ROC, and F1-score**.
4. Connect **Performance** output to **Results** and run the process to view model performance.

Repository

- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)

Operators

read

- Data Access (27)
 - Files (14)
 - Read (13)
 - Read CSV
 - Read Excel
 - Read URL
 - Read Access
 - Read SPSS
 - Read Stata
 - Read ARFF

We found "SharePoint Connector", "OPC-UA Connector" and 7 more results in the Marketplace. [Show me!](#)

Process

Process

110%

Parameters

Performance (3) (Performance (Binominal Classification))

- ☒ accuracy
- ☐ classification error
- ☐ kappa
- ☐ AUC (optimistic)
- ☐ AUC
- ☐ AUC (pessimistic)
- ☒ precision
- ☒ recall
- ☐ lift

[Hide advanced parameters](#)

Help

Performance (Binominal Classification)

AI Studio Core

Tags: Roc, Accuracy, Errors, Precision, Recall, Auc, Lift, True, False, Positives, Negatives, Metrics, Confusion Matrix, Predictive

Synopsis

This Operator is used to statistically evaluate the strengths and weaknesses of a binary classification, after a trained model has been applied to labelled data.

[Jump to Tutorial Process](#)

Decision Tree Performance:

PerformanceVector (Performance (3))

Result History

Criterion

- accuracy
- precision
- recall
- f measure

Performance

Description

Annotations

PerformanceVector (Performance (4))

Improved Neural Net (Neural Net)

Tree (Decision Tree)

Kernel Model (SVM)

PerformanceVector (Performance)

Table View Plot View

accuracy: 70.87%

	true false	true true	class precision
pred. false	120	37	76.43%
pred. true	30	43	58.90%
class recall	80.00%	53.75%	

Repository

- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)

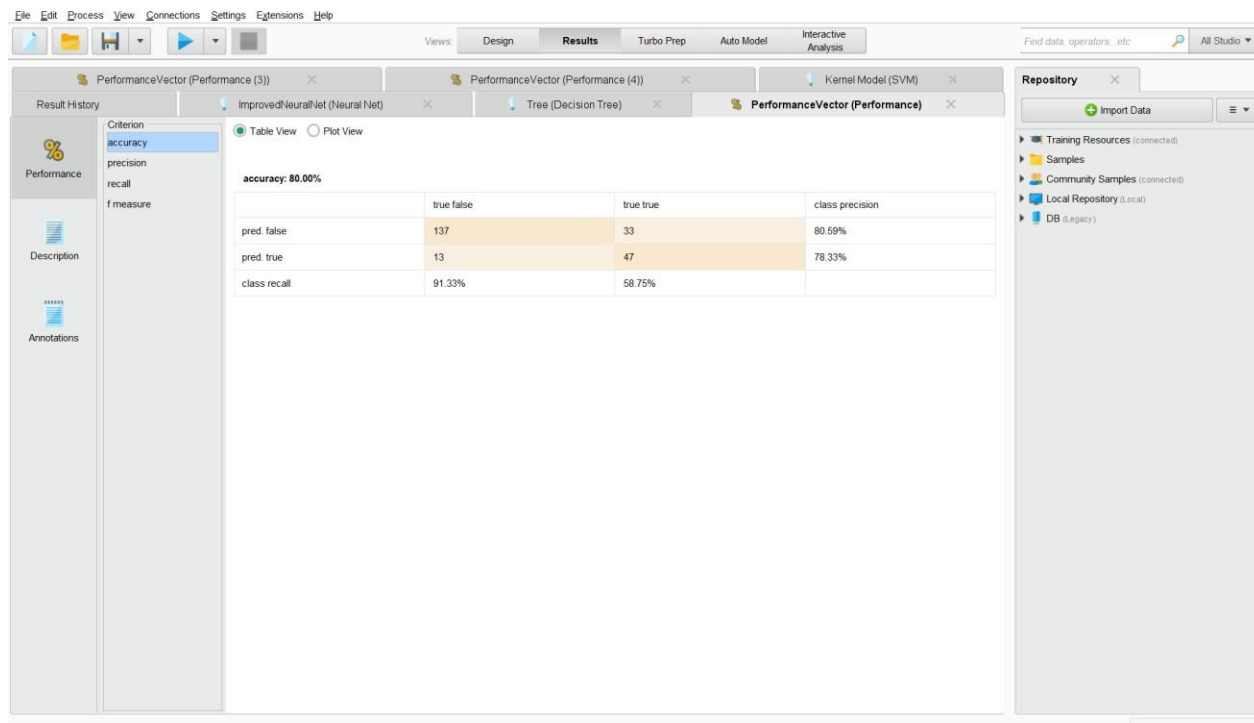
Accuracy→ 70.87%

Precision→ 58.90%

Recall→ 53.75%

F_measure → 56.21%

SVM Performance:



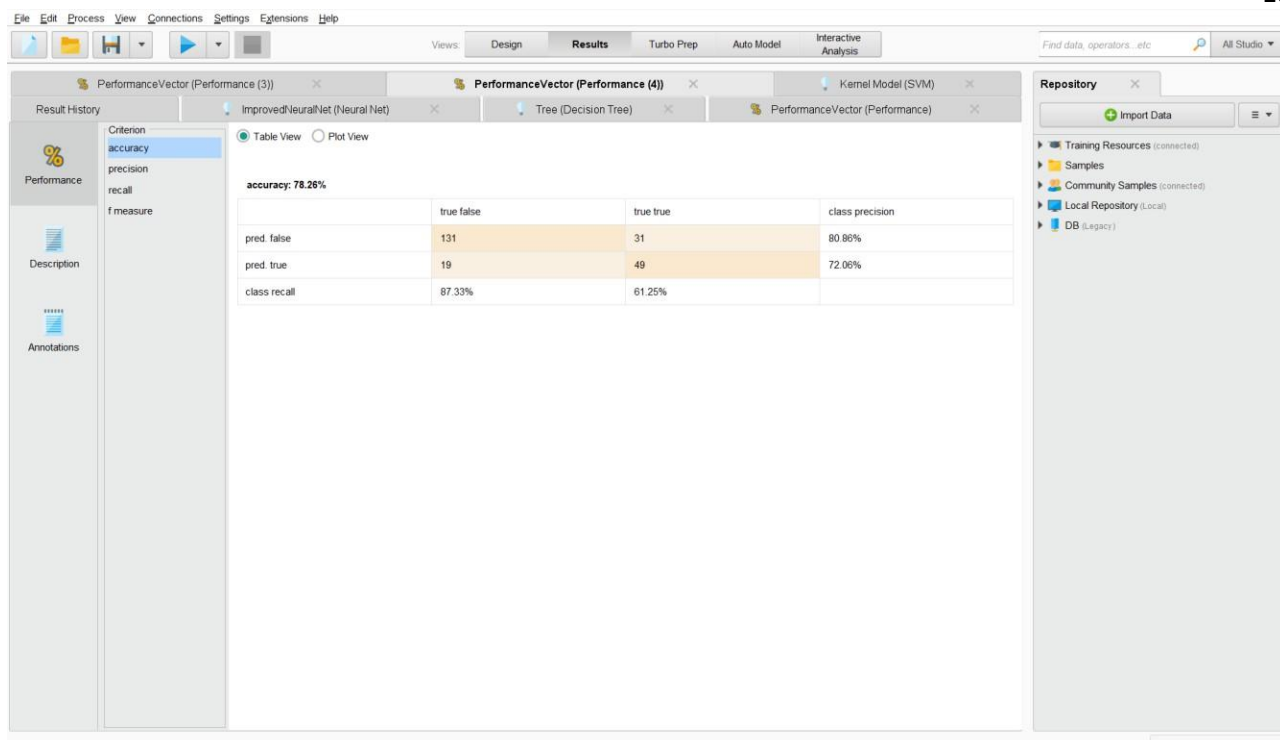
Accuracy→ 80.0%

Precision→ 78.33%

Recall→ 58.75%

F_measure → 67.14%

Neural Network Performance:



Accuracy → 78.26%

Precision → 72.06%

Recall → 61.25%

F_measure → 66.22%

Result:

After performing **numerical to binominal conversion, replacing missing values, normalizing data, selecting important features, splitting data, training models, and evaluating performance**, the final model achieves an accuracy of **70%+**, providing an effective solution for predicting diabetes risk.

Outcome:

	Decision Tree	SVM	Neural Network
Accuracy	70.87	80.0	78.26
Precision	58.90	78.33	72.06
Recall	53.75	58.75	61.25
F_measure	56.21	67.14	66.22

S