
# AIML Project Report

# Heart Failure Prediction Using KNIME Workflow

Department of Electronics and Communication Engineering, 2024-25

**Done by Student:** 23ECB0A30 – Sambaraju Sree Vardhan

**Under the supervision of:** Professor Dr. K. Ravi Kishore, Department of ECE, NIT Warangal

# Introduction:

The goal of this project is to predict the risk of heart failure in patients using a machine learning model built in KNIME. By analysing medical data, this project aims to identify patients at higher risk of heart failure, which could allow for earlier interventions and better treatment outcomes. KNIME Analytics Platform is an open-source software that allows users to access, blend, analyse, and visualize data, without any coding. Its low-code, no-code interface offers an easy introduction for beginners, and an advanced data science set of tools for experienced users.

# About the Dataset:

The dataset used in this project includes patient information, such as age, sex, blood pressure, cholesterol levels, and other health-related metrics. Key attributes are:

1. Age: The age of the patient in years.

2. Anaemia: Indicates if the patient has anaemia, a condition marked by a lower-than-normal count of red blood cells.

3. Creatinine Phosphokinase: Measures the level of an enzyme in the blood, often associated with heart muscle damage.

4. Diabetes: Indicates if the patient has diabetes.

5. Ejection Fraction: The percentage of blood that leaves the heart with each contraction, an important metric in assessing heart function.

6. High Blood Pressure: Indicates if the patient has high blood pressure.

Platelets: The count of platelets in the blood, which plays a role in clotting.

7. Serum Creatinine: Measures the level of creatinine in the blood, which helps assess kidney function.

8. Serum Sodium: The level of sodium in the blood, which can impact heart function.

9. Sex: The gender of the patient, where 1 represents male and 0 represents female.

10.Smoking: Indicates if the patient is a smoker.

11.Time: The period of observation for each patient, usually in days.

12.DEATH_EVENT: The target class or outcome, where 1 indicates the patient experienced heart failure, and 0 indicates a normal outcome.

## Nodes and their functions:

CSV Reader Node:

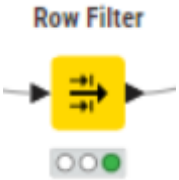

Input - CSV file containing the dataset.

Output - Structured table with the dataset in rows and columns format, ready for processing

- Reads the dataset from a CSV file format and converts it into a structured table for processing

- Prepares the data for processing by reading and organizing it into a structured table that can be used in subsequent steps.

Row Filter Node:




Input - Structured table from the CSV Reader.

Output - Filtered table with rows meeting specified conditions (e.g., no incomplete or erroneous data).

- Filters out irrelevant, incomplete, or erroneous rows based on specified conditions.

- Ensures only clean and valid data is used for analysis and model training.

Missing Value Node:

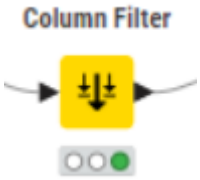

Input - Filtered table from the Row Filter node.

Output - Table with missing values handled, either by filling them or removing rows/columns with missing data.

- Identifies and handles missing values in the dataset.

- Fills or removes missing data to maintain dataset integrity before training the model.

Column Filter Node:



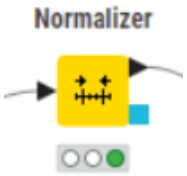

Input - Table with missing values handled from the Missing Value node.

Output - Table with only the relevant columns selected for analysis.

- Selects only the relevant columns required for analysis.

- Simplifies the dataset by eliminating unnecessary or redundant features.

Normalizer Node:



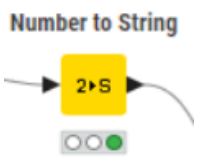

Input - Filtered table from the Column Filter node.

Output - Normalized table with numeric features scaled to a common range or standardized for consistent model input.

- Scales numeric features to a common range or standardizes them.

- Ensures that features are comparable, improving model performance and convergence.

Number to String Node:



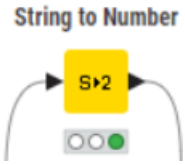
Input - Normalized table from the Normalizer node.

Output - Table with specified numeric columns converted to string format for text-based analysis or specific processing requirements.

- Converts numerical values in selected columns to string format.

- Useful for reformatting data for text-based analysis or specific processing requirements.

String to Number Node:



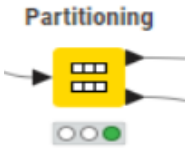

Input - Table with certain columns in string format from the Number to String node.

Output - Table with specified string columns converted back to numerical format, ensuring compatibility for model training.

- Converts string data back into numerical format where applicable.

- Ensures numeric columns are in the correct format for model training or statistical analysis.

Partitioning Node:




Input - Preprocessed table with columns in the correct format.

Output - Split dataset, with one subset for training (e.g., 80%) and another for testing (e.g., 20%).

- Splits the dataset into training and testing sets (80-20 split).

- This node divides the dataset into two (or more) subsets: one for training the model and one for testing it. The partitioning can be done randomly or based on specified proportions (e.g., 80% for training and 20% for testing).

Decision Tree Learner Node:


Input - Training subset from the Partitioning node.

Output - Trained decision tree model based on the training data.

- Trains a decision tree model on the training dataset.

- This node applies the decision tree algorithm to the training data. It recursively splits the data based on the best features, according to criteria like Gini impurity or Information Gain, to build a tree structure. The tree classifies instances by making decisions at each node based on feature values. The learner node will output a trained model that can be used for prediction.

Decision Tree Predictor Node:

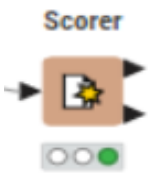

Input - Trained model from the Decision Tree Learner and test subset from the Partitioning node.

Output - Predicted labels for the test data, assigning each instance to a class (e.g., "at risk" or "not at risk").

- Uses the trained decision tree model to make predictions on the testing set.

- After training the decision tree model, this node takes the test data (unseen during training) and uses the learned decision rules to predict outcomes (e.g., whether a patient is at risk of heart failure). It assigns each instance to a class (e.g., "at risk" or "not at risk") based on the feature values and decision criteria established during the training phase.

Scorer Node:

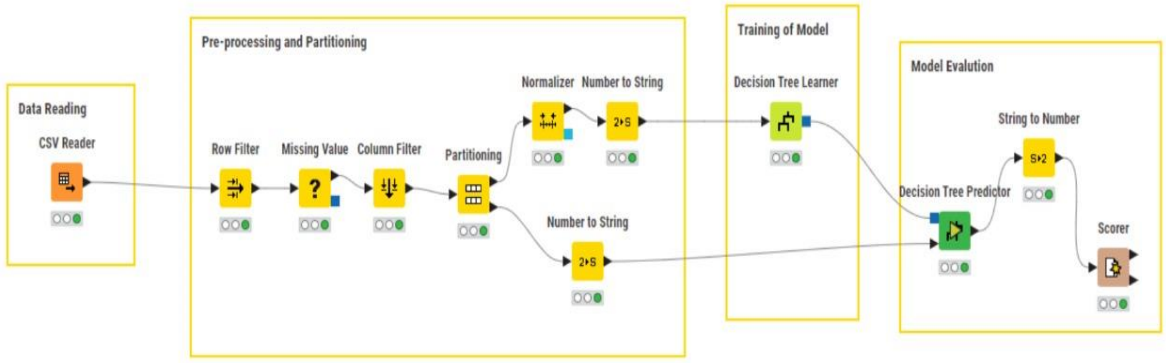



Input - Predicted labels from the Decision Tree Predictor and true labels from the test subset.

Output - Performance metrics (e.g., accuracy, precision, recall, F1-score, sensitivity, specificity) assessing the model's performance.

- Evaluates model predictions against the true labels using various metrics.

- The Scorer node calculates several performance metrics based on the model's predictions, including accuracy, precision, recall, F1-score, sensitivity, and specificity. These metrics are used to assess how well the model is performing in terms of correctly identifying heart failure risk. For example, accuracy shows the overall success rate, while precision and recall give insight into how well the model handles false positives and false negatives, respectively.




**Knime Workflow for Heart Failure Prediction**

# Accuracy Statistics:

Columns: 11

| RowID | TruePositives Number (Integer) | FalsePositives Number (Integer) | TrueNegatives Number (Integer) | FalseNegatives Number (Integer) | Recall Number (double) | Precision Number (double) | Sensitivity Number (double) | Specificity Number (double) | F-measure Number (double) | Accuracy Number (double) |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.0 | 44 | 16 | 0 | 0 | 1 | 0.733 | 1 | 0 | 0.846 | ? |
| 1.0 | 0 | 0 | 44 | 16 | 0 | ? | 0 | 1 | ? | ? |
| Over... | ? | ? | ? | ? | ? | ? | ? | ? | ? | 0.733 |

Overall Accuracy: 73.3%

Precision (No Heart Failure): 73.3%

Recall (No Heart Failure): 100%

F1-Score (No Heart Failure): 84.6%

## Confusion Matrix:

| RowID | 0.0 *Number (integer)* | 1.0 *Number (integer)* |
|---|---|---|
| 0.0 | 44 | 0 |
| 1.0 | 16 | 0 |

## Conclusion:

The model shows promise in identifying heart failure risks, supporting early detection and preventive care efforts. While it performs reliably in distinguishing patients, enhancing its ability to identify those truly at risk would make it even more effective. By improving recall, the model could help ensure that more high-risk patients receive timely attention, ultimately contributing to better health outcomes and proactive management of heart conditions.