KNIME Workshop Report on Machine Learning Algorithms

M.Masuod

June 9, 2024

Abstract

Executive Summary: This report summarizes the activities and outcomes of the KNIME workshop on Machine Learning Algorithms. The workshop aimed to familiarize participants with the KNIME analytics platform and to understand and apply various machine learning algorithms, including K-means clustering, decision trees, and neural networks. The hands-on approach ensured practical experience in implementing these algorithms using the KNIME platform.

Contents

1. KNIME and Machine Learning

1.1 Introduction

The aim of this experiment was to familiarize participants with the KNIME analytics platform and to understand and apply various machine learning algorithms, including K-means clustering, decision trees, and neural networks. The workshop's hands-on approach ensured that participants not only learned the theoretical aspects of these algorithms but also gained practical experience in implementing them using the KNIME platform.

1.2 Aim/Objectives

The main objectives of the workshop were:

- To understand and apply various machine learning algorithms using the KNIME platform.
- To gain hands-on experience in implementing K-means clustering, decision trees, and neural networks.
- To learn data preprocessing, model training, and evaluation techniques.

1.3 Overview of Systems/Programs/Models

1.3.1 KNIME Analytics Platform

KNIME (Konstanz Information Miner) is an open-source data analytics, reporting, and integration platform. It supports a wide range of machine learning and data mining algorithms, which can be integrated using a graphical user interface that allows users to visually create data flows by connecting different modules. KNIME supports plugins written in Java, Python, and R, and includes modules for data integration from various databases such as SQL, MySQL, PostgreSQL, and MongoDB.

1.3.2 Machine Learning Algorithms

K-means Clustering

Concept: A clustering algorithm that partitions a dataset into K clusters by assigning data points to the nearest cluster centroids.
Procedure: Initialization, centroid calculation, data point reallocation, and iteration until no further reallocation.

Decision Trees

Concept: A classification model that uses a tree-like graph of decisions and their possible consequences.
Procedure: Create nodes and leaves based on attribute values, construct sub-trees, and connect them to form the complete decision tree.

Neural Networks

Concept: Computational models inspired by the human brain, consisting of interconnected neurons that process input data and generate outputs.
Procedure: Data normalization, network partitioning, training with a backpropagation algorithm, and prediction using the trained model.

1.4 Results Obtained According to the Procedure in the Workshop

1.4.1 K-means Clustering

Data Reading: Successfully read data from a CSV file using the File Reader module.
Clustering Execution: Applied K-means clustering with different numbers of clusters (2, 3, and 4).

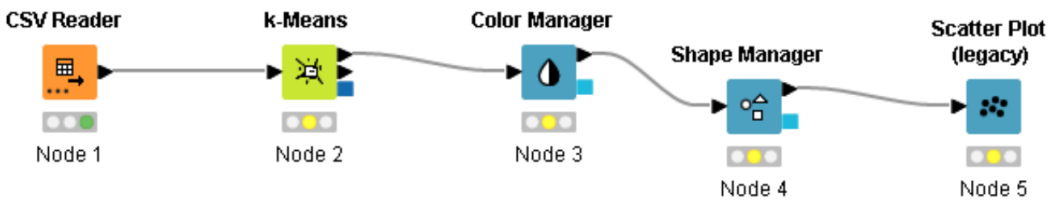


Figure 1.1: Schematic for K-means Clustering

Results: The number of clusters was determined by observing the scatter plots generated for different cluster values and using the Elbow method for validation.

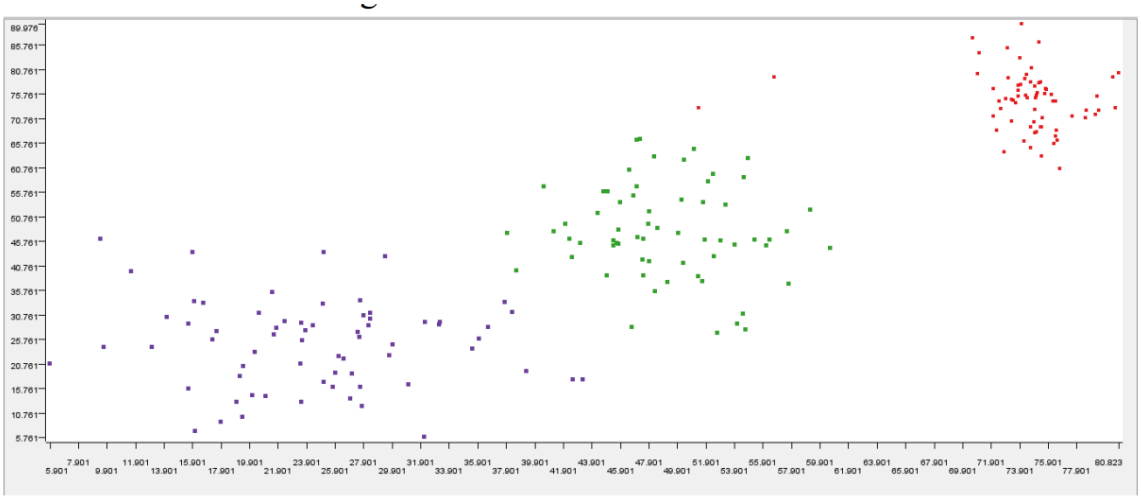


Figure 1.2: Cluster View

Questions:

- **Q1:** What will happen if you check "read row IDs"?
- **A1:** It will consider the first column as the row number in the data file.
- **Q2:** How many rows are there?
- **A2:** There are 198 rows in the data excluding the one with the column header.
- **Q3:** According to your results, how many clusters are there?
- **A3:** Based on the results from the elbow method, the optimal number of clusters is 3. The elbow method involves plotting the explained variance as a function of the number of clusters and picking the point where the variance starts diminishing, which forms an "elbow" shape. In this case, the elbow occurs at 3 clusters, indicating it as the best fit for the input data.

1.4.2 Decision Trees

Data Partitioning: Divided data into training and testing sets (80% and 20% respectively).
Model Training: Trained a decision tree model using the training data.

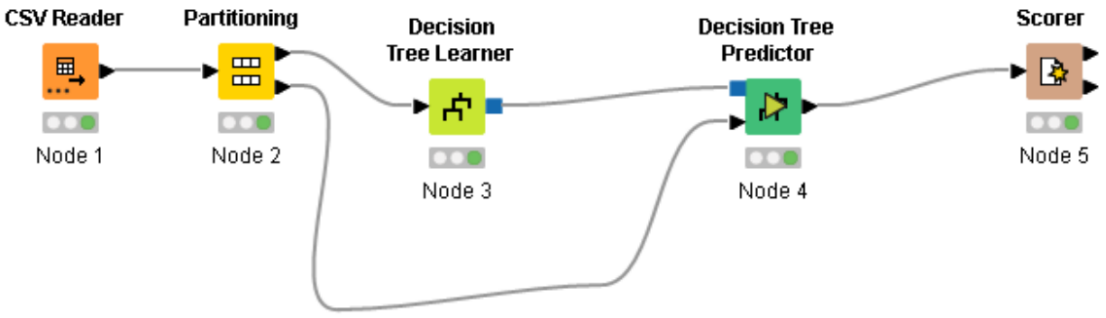


Figure 1.3: Schematic for Decision Tree

Evaluation: Evaluated the model using the testing data, generating a confusion matrix to assess performance.

File Hilite

income \ Pr...	<=50K	>50K
<=50K	4550	470
>50K	558	935

Figure 1.4: Confusion Matrix of Outputs

Questions:

- **Q1:** How many rows are there in the first partition? How about the second partition?
- **A1:** There are 26048 rows in the first partition and 6513 rows in the second partition of the provided data.
- **Q2:** Which attribute has been chosen as the root of the tree?
- **A2:** The attribute chosen as the root of the tree is "capital-gain".
- **Q3:** What does the confusion matrix tell you?
- **A3:** The confusion matrix provides the following insights about the performance of the decision tree model:
  - **True Positives (TP):** The number of instances correctly predicted as "¿50K" is 935.
  - **True Negatives (TN):** The number of instances correctly predicted as "¡=50K" is 4550.
  - **False Positives (FP):** The number of instances incorrectly predicted as "¿50K" is 470.
  - **False Negatives (FN):** The number of instances incorrectly predicted as "¡=50K" is 558.

  This matrix helps in understanding the accuracy and error distribution of the model's predictions. The high values of TP and TN compared to FP and FN indicate a generally good performance.

1.4.3 Neural Networks

Data Preparation: Normalized data before feeding it into the neural network.
Model Training: Trained a neural network using the RProp MLP Learner with specified parameters.

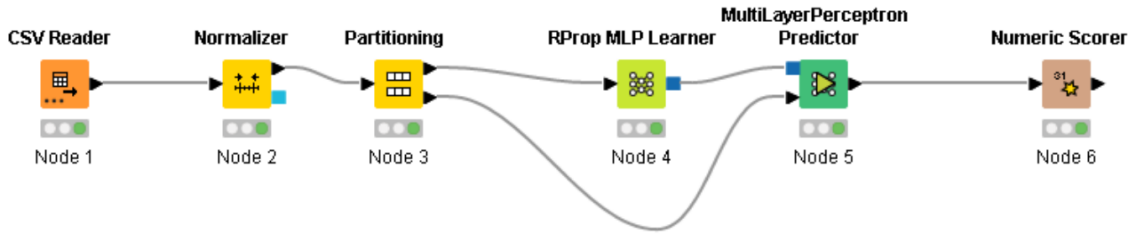


Figure 1.5: Schematic for Neural Network

Model Evaluation: Assessed the accuracy of the model by observing the effects of iterations, hidden layers, and neurons per layer on the model's performance.
Questions:

R^2:	1
Mean absolute error:	0.001
Mean squared error:	0
Root mean squared error:	0.002
Mean signed difference:	-0
Mean absolute percentage error:	1.481
Adjusted R^2:	1

Figure 1.6: Best Model Results

- **Q:** What are the effects of the numbers of a) iterations, b) hidden layers, and c) hidden neurons per layer on the accuracy of the model?
- **A:** Based on the observed trends:
  - **Iterations:** Increasing the number of iterations consistently reduces the Mean Absolute Percentage Error (MAPE). More iterations allow the model to better learn from the data, improving accuracy.
  - **Hidden Layers:** Increasing the number of hidden layers reduces MAPE up to a certain point (specifically, up to 5 layers). Beyond this, adding more layers starts to increase the error, suggesting that up to 5 layers help the model capture more complex patterns in the data, but beyond that, the model becomes too complex and starts overfitting or becoming less efficient.
  - **Hidden Neurons per Layer:** Increasing the number of neurons per layer decreases MAPE up to 10 neurons per layer. Beyond 10 neurons, the error starts to increase, indicating that 10 neurons per layer strike an optimal balance between model complexity and performance. Adding more neurons beyond this point likely leads to overfitting or unnecessary complexity, reducing the model's ability to generalize.

  In summary:

  - **Iterations:** More iterations improve accuracy (reduce MAPE).
  - **Hidden Layers:** Up to 5 layers improve accuracy, but more than 5 layers increase error.
  - **Hidden Neurons per Layer:** Up to 10 neurons per layer improve accuracy, but more than 10 neurons increase error.

  These insights help in understanding how to adjust these parameters to achieve optimal model performance.

1.5 Discussion and Conclusions

1.5.1 Discussion

The workshop provided a comprehensive understanding of implementing machine learning algorithms using the KNIME platform. The hands-on exercises helped in solidifying the theoretical knowledge by applying it to real-world data.

- **K-means Clustering:**
  - Clustering results varied with the number of clusters. The Elbow method was effective in determining the optimal number of clusters.
  - Visual representation using scatter plots facilitated the understanding of data distribution across clusters.
- **Decision Trees:**
  - The decision tree model efficiently classified the data based on the chosen attributes.
  - The confusion matrix was a crucial tool in evaluating the model's accuracy and identifying areas for improvement.
- **Neural Networks:**
  - Normalization significantly impacted the performance of the neural network, ensuring consistent input data scaling.
  - Increasing the number of iterations, hidden layers, and neurons generally improved the model's accuracy but also increased computational complexity.

1.5.2 Conclusions

The KNIME workshop successfully demonstrated the implementation and application of K-means clustering, decision trees, and neural networks. Participants gained valuable skills in data preprocessing, model training, and evaluation using KNIME's user-friendly interface. The hands-on approach of the workshop ensured that theoretical concepts were well-understood and could be applied practically, providing a strong foundation for further exploration in machine learning and data analytics.

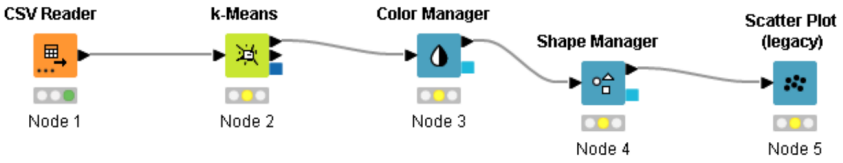
.1 Appendices

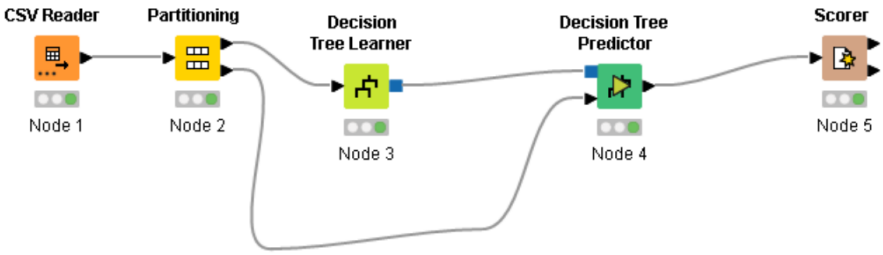
.1.1 Appendix A: KNIME Workflows

Detailed KNIME workflow configurations and screenshots.

K-means Clustering Workflow

- Workflow Name: KMeans-Clustering
- Description: This workflow demonstrates the implementation of above tasks in KNIME.
- Graphics:

1. 

2. 

.1.2 Appendix B: Dataset Specifications

Specifications and descriptions of the datasets used in the workshop.

Dataset 1: Customer Segmentation

- Source: University Portal
- Description: This dataset contains information about customers, including age, income, and spending score. It is used for customer segmentation analysis.
- Format: CSV

A. References

- https://www.knime.com/knime-analytics-platform
- https://www.knime.com/documentation