ሀዋሳ ዩኒቨርሲቲ
HAWASSA UNIVERSITY

Faculty of Informatics

Department of Information Systems

Introduction to Datamining and Machine Learning (InSy3051)

Thyroid Recurrence Prediction

1. Tsion Tesfaye	**NaScR/3725/16**
2. Biniyam Fisseha	**NaScR/1609/16**
3. Afomiya Million	**NaScR/1199/16**
4. Yonatan Berihun	**NaScR/3962/16**
5. Elham Seid	**NaScR/1924/16**
6. Amanuel Wondmagegnehu	**NaScR/1274/16**
7. Fenet Bushura	**NaScR/2105/16**
8. Abdulkerim Kedir	**NaScR/1027/16**

Submitted to: Ms. Kalkidan

Submission Date: December 15, 2025

Thyroid Recurrence Prediction

1. Introduction

The objective of this project is to build and compare machine learning models to predict whether a thyroid condition will recur in a patient. The dataset filtered_thyroid_data.csv contains patient records with various attributes such as age, gender, pathology, and treatment history. We implemented two classification algorithms to analyze the data and predict recurrence outcomes:

- **Decision Trees** and
- **K-Nearest Neighbors (KNN)**

2. Data Preprocessing

Before training the models, the raw data underwent several preprocessing steps to ensure quality and compatibility with the machine learning algorithms:

- **Missing Value Handling:** The dataset contained missing values in several categorical columns. These were imputed using the **mode** (most frequent value) of the respective columns to preserve the natural distribution of the data.
- **Target Encoding:** The target variable, Recurred, was categorical (Yes/No). It was encoded into a numerical format (1 for Yes, 0 for No) using Label Encoding to make it compatible with the models.
- **Feature Encoding:** Categorical features (e.g., Risk, Stage, Pathology, Focality) were converted into numerical form using **One-Hot Encoding**. This created binary dummy variables for each category (e.g., Stage_I, Stage_II), preventing the models from assuming a false ordinal relationship between nominal categories.
- **Feature Scaling:** We applied **StandardScaler** to normalize the feature set, transforming data to have a mean of 0 and a standard deviation of 1. This step was particularly critical for the KNN algorithm, which relies on Euclidean distance and is otherwise sensitive to the magnitude of features

(e.g., Age vs. binary flags).
- **Data Splitting:** The data was split into training (80%) and testing (20%) sets to evaluate model performance on unseen data.

3. Methodology

3.1 Decision Tree Classifier

- **Description:** The Decision Tree is a non-parametric supervised learning method used for classification. It predicts the target by learning simple decision rules inferred from the data features. It splits the data into subsets based on the most significant attribute at each node.
- **Configuration:** We used the standard DecisionTreeClassifier from the Scikit-Learn library with a random state set for reproducibility.

3.2 K-Nearest Neighbors (KNN)

- **Description:** KNN is an instance-based learning algorithm where a new data point is classified based on the majority class of its 'k' nearest neighbors in the feature space. It assumes that similar things exist in close proximity.
- **Configuration:** We configured the model with n_neighbors=5 and used the standard Euclidean distance metric.

4. Results and Comparison

The models were evaluated on the test set (20% of the data). The performance metrics are summarized below:

Metric	Decision Tree	KNN (K=5)
Accuracy	**97.40%**	94.81%
Precision (Recurred=1)	0.95	1.00
Recall (Recurred=1)	0.95	0.79
F1-Score	0.95	0.88

Observations:

- **Decision Tree Performance:** The Decision Tree achieved a higher overall accuracy of **97.4%**. Notably, it provided balanced performance in identifying both recurred and non-recurred cases, with a recall of **0.95** for the positive class. This means it successfully identified 95% of the patients who actually had a recurrence.
- **KNN Performance:** The KNN model achieved an accuracy of **94.8%**. While it was extremely precise (Precision of **1.00**, meaning every patient it predicted as "Recurred" did indeed recur), it suffered from lower recall (**0.79**). It missed approximately 21% of the actual recurrence cases.

5. Conclusion

Based on the comparison, the **Decision Tree** model proved to be the superior classifier for this specific dataset.

While KNN offered perfect precision, the Decision Tree's ability to capture 95% of recurrence cases (Recall) makes it more suitable for a medical diagnostic context where missing a positive case (false negative) is a critical error. The Decision Tree likely handled the high dimensionality introduced by One-Hot Encoding better than the distance-based KNN algorithm.