Project Report - Subject 5 : Thyroid Cancer Recurrence

MARTINETTI Paul, MERIC Axelle, VAST Julien,
SFAR Mehdi, LAFRENIERE Juliette, BRAUN Baptiste

June 6, 2025

Contents

1 Introduction

As indicated by its title, the aim of this project is to study the recurrence of thyroid cancer. Identifying risk factors for thyroid cancer recurrence, especially in relation to the specific type of cancer, could help guide physicians in two key areas:

- Choosing the most appropriate treatment strategy at the time of diagnosis;
- Defining the optimal follow-up protocol during and after treatment.

1.1 Dataset Presentation

In this project we use a dataset from [1]. It consists of 383 patients who had previously had thyroid cancer. There is exactly one continuous covariate, which is `Age`. The 15 other covariates are categorical variables, as shown in Figure 1. The target variable is `Recurred`, which indicates whether or not the cancer recurred for each patient. The problem can be formulated as a supervised binary classification task, where the goal is to predict the recurrence of thyroid cancer using a set of clinical and pathological covariates.

Table 1: Description and modalities of the covariates (features)

Feature (Covariate)	**Modalities**
`Age`	Continuous variable (e.g., 15 to 82 years)
`Gender`	F (Female), M (Male)
`Smoking`	No, Yes
`Hx Smoking`	No, Yes
`Hx Radiotherapy`	No, Yes
`Thyroid Function`	Euthyroid, Clinical Hyperthyroidism, Clinical Hypothyroidism, Subclinical Hyperthyroidism, Subclinical Hypothyroidism
`Physical Examination`	Normal, Diffuse goiter, Single nodular goiter-left, Single nodular goiter-right, Multinodular goiter
`Adenopathy`	No, Right, Left, Bilateral, Posterior, Extensive
`Pathology`	Papillary, Micropapillary, Follicular, Hurthle cell
`Focality`	Uni-Focal, Multi-Focal
`Risk`	Low, Intermediate, High
`T`	T1a, T1b, T2, T3a, T3b, T4a, T4b
`N`	N0, N1a, N1b
`M`	M0, M1
`Stage`	I, II, III, IVA, IVB
`Response`	Excellent, Indeterminate, Biochemical Incomplete, Structural Incomplete

Table 2: Description of the target variable

Target Variable	**Description**
`Recurred`	Indicates whether the cancer recurred after initial treatment (No, Yes). This is the variable to be predicted.

In this project, we will compute the number of false positives and false negatives obtained for each model evaluated.

In this medical context, a false positive means classifying a patient as being at risk of recurrence when they are not. In such a case, the patient would benefit from enhanced follow-up care, which is generally beneficial. However, they may also receive a more aggressive treatment that could lead to unnecessary side effects,,which could be avoided if the patient had been correctly identified.

Conversely, a false negative means failing to identify a patient who is truly at risk of recurrence. This patient would not receive the intensive treatment or monitoring they should have, resulting in a significant loss of opportunity for better outcomes. Given that false negatives carry more serious clinical consequences than false positives, our objective throughout this project is not only to maintain strong precision, but also to keep the number of false negatives as low as possible.

1.2 Analysis of the variables

Before building any model, the first thing to do is look at the variable and the dataset where our target variable is the `Recurred` with two modalities `yes/no`. The variable indicates if a patient who had been cured from cancer have had cancer back or not.

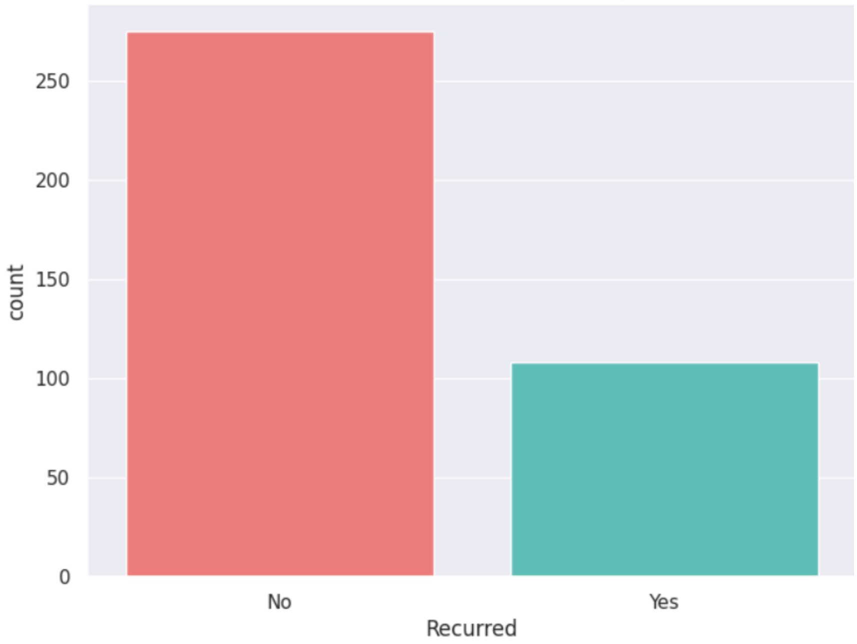


Figure 1: Distribution of `Recurred` in Training Set

In this study, the dataset is unbalanced yet there are no missing data. We also notice that some lines can be identical. We will consider that it corresponds to different people whom share the same answers to the variables of this dataset as it is highly probable to occur in such situations. We focus this brief analysis on four variables which seem to be the most interesting ones.

1.2.1 Age

The `Age` variable ranges from 15 to 82, with a mean of 41 and a median of 37. The histogram shows that most patients cluster around 30, with a slight tail toward higher values.

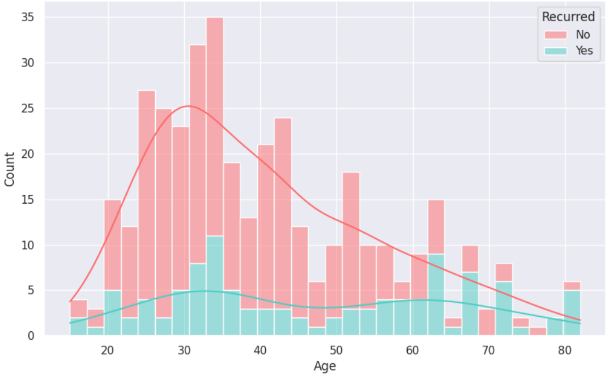


Figure 2: Distribution of `Age` in Training Set

1.2.2 Gender

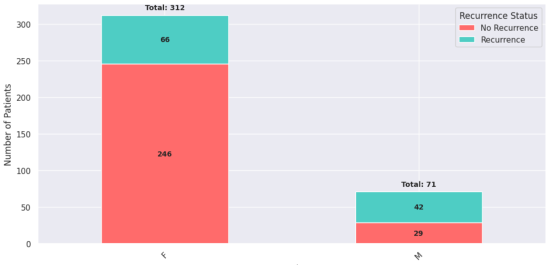
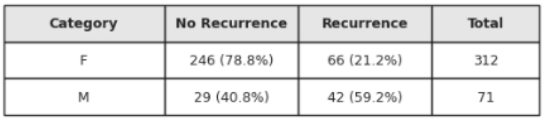


Figure 3: Distribution of `Gender` by Recurrence Status



Category	No Recurrence	Recurrence	Total
F	246 (78.8%)	66 (21.2%)	312
M	29 (40.8%)	42 (59.2%)	71

Figure 4: Proportion of the variable `Gender`

Out of a total of 383 patients, 312 (81.5 %) are female and 71 (18.5 %) are male. The bar chart clearly illustrates the predominance of women, which is consistent with the higher prevalence of thyroid diseases among females.

1.2.3 Distribution T

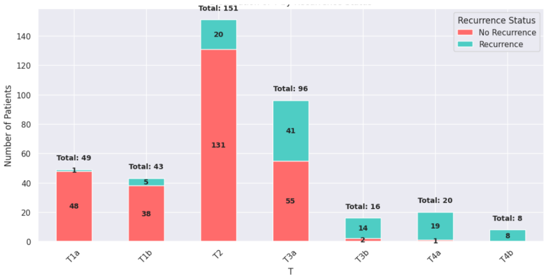
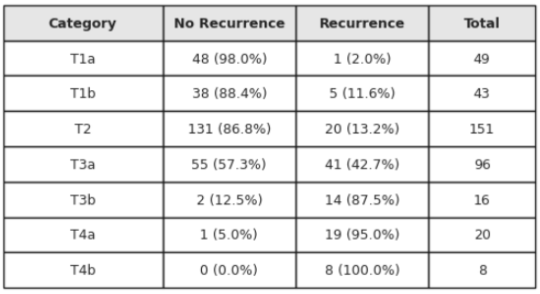


Figure 5: Distribution of `T` by Recurrence Status



Category	No Recurrence	Recurrence	Total
T1a	48 (98.0%)	1 (2.0%)	49
T1b	38 (88.4%)	5 (11.6%)	43
T2	131 (86.8%)	20 (13.2%)	151
T3a	55 (57.3%)	41 (42.7%)	96
T3b	2 (12.5%)	14 (87.5%)	16
T4a	1 (5.0%)	19 (95.0%)	20
T4b	0 (0.0%)	8 (100.0%)	8

Figure 6: Proportion of the variable `T`

The `T` variable reflects tumor size, with categories on the left indicating less severe stages and those on the right representing more advanced, severe stages. The distribution of `T` stages shows that the most frequent category is T2 (151 patients, 39.4 %). This is followed by T3a (96 patients, 25.1 %) and T1a (49 patients, 12.8 %). The more advanced local stages (T3a, T3b, T4a, T4b) together represent 36.6 % of the cohort, indicating that a substantial proportion of diagnoses occurred at a more advanced local stage. Stages T3b, T4a, and T4b are relatively rare (11.5 %), suggesting that truly late-stage cases are less common in this sample.

1.2.4 Response

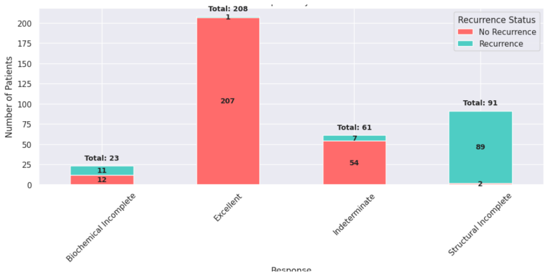
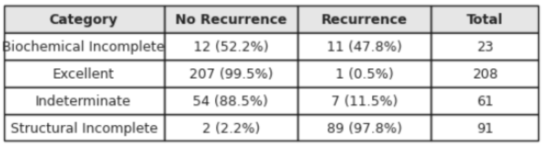


Figure 7: Distribution of Response by Recurrence Status



Category	No Recurrence	Recurrence	Total
Biochemical Incomplete	12 (52.2%)	11 (47.8%)	23
Excellent	207 (99.5%)	1 (0.5%)	208
Indeterminate	54 (88.5%)	7 (11.5%)	61
Structural Incomplete	2 (2.2%)	89 (97.8%)	91

Figure 8: Proportion of the variable Response

Among the patients followed , 208 (54.3 %) show an “Excellent” response, 23 (6.0 %) are “Biochemical Incomplete,” 91 (23.8 %) are “Structural Incomplete,” and 61 (15.9 %) remain “Indeterminate.” The “Excellent” category dominates the distribution, but nearly 29.8 % of patients have an incomplete response (biochemical or structural), indicating the need for closer follow-up in this subgroup. Although the responses "Excellent", "Indeterminate", and "Structural Incomplete" are all represented, a clear trend can be observed: if a patient has a "Excellent" or "Indeterminate" response, they are very unlikely to experience a recurrence. Conversely, if the patient has a "Structural Incomplete" response, they are very likely to experience a recurrence.

2 Pre-processing

During the pre-processing stage, for all used methods (except Random Forest), we use one-hot encoding for all covariates with more than two categories. An example of this encoding method is presented below for the covariate Risk:

Table 3: Example of one-hot encoding for the variable Risk

Original Category	Risk_Low	Risk_Intermediate	Risk_High
Low	1	0	0
Intermediate	0	1	0
High	0	0	1

For binary categorical covariates, we simply encode them using 0 and 1, which preserves a single column per variable.

Table 4: Binary encoding of categorical covariates

Variable	**0**	**1**
Gender	Female (F)	Male (M)
Smoking	No	Yes
Hx_Smoking	No	Yes
Hx_Radiotherapy	No	Yes
Focality	Multi-Focal	Uni-Focal
M	M0	M1
Recurred	No	Yes

Since Age is the only continuous covariate, it was standardized in order to have zero mean and unit variance, which helps prevent it from disproportionately influencing models that are sensitive to the scale of input features.

3 Choice of Metrics

To compare our models, we will use the following metrics:

- **Accuracy:** the proportion of total predictions (both positive and negative) that are correct. It is defined as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

- **Recall:** the proportion of actual positive cases that are correctly identified. It is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **AUC (Area Under the ROC Curve):** a metric that measures the model's ability to distinguish between classes. AUC represents the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative one. The ROC curve is the parametric curve which associated to the set

$$\{(FPR(t), TPR(t)), \text{for all threshold t}\}$$

  where TPR(t) is the recall and FPR(t) is the false positive rate when t is the threshold.

$$\left(\text{False Positive Rate } = \frac{\text{False Positives}}{\text{True Negatives} + \text{False Positives}}\right)$$

  and the AUC is the area under this curve. Since the classes are unbalanced, the accuracy can be misleading due to the under representation of one class. This is why we also compute the AUC, with the goal of achieving the highest possible AUC value which allows us to know if the model has sufficient discriminatory power to make it worth choosing a threshold.

Sometimes, we will consider an other metric, the F1-score as the following metric:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

where the *precision* is the proportion of positive predictions that are actually correct. It is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

The F1-score allows us to avoid one of precision or recall to not be adequate and to create a compromise between these two metrics.
The F$\boldsymbol{\beta}$-score is a generalized version of the F1-score that allows adjusting the relative importance of recall versus precision. It is particularly useful in medical applications where false negatives may have more serious consequences than false positives. The formula for this metric is:

$$\boldsymbol{F_\beta = (1 + \beta^2)\,\frac{\textbf{Precision} \times \textbf{Recall}}{\beta^2 \times \textbf{Precision} + \textbf{Recall}}.}$$

Given the clinical importance of minimizing false negatives, particular attention will be paid to maximizing recall, while also seeking to achieve high precision and AUC.

4 Model Evaluation

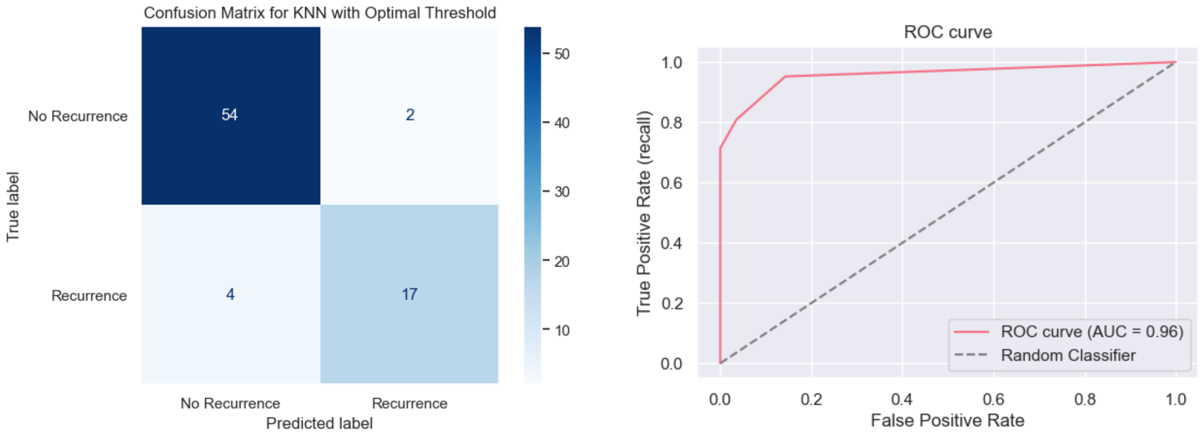
In this section, we will list the machine learning methods used to solve this problem, explaining their principles if they have not been covered in class, and giving some advantages and drawbacks. The performance results will be given in the next section. For each method, we split our dataset into three parts: 60% for training the model, 20% for validation (used to optimize the classification threshold and, in the case of the DNN, to select a suitable architecture), and 20% for testing.
Optimizing the threshold helps reducing the number of false positives as much as possible, in agreement with our purpose of improving performance and increasing the recall.

4.1 Model 1: [K Nearest Neighbors (KNN)]

During model training, we use 5-fold cross-validation to select the best parameters (the choice of the metric and the number of neighbors) from a well-chosen grid of candidate values.

4.1.1 KNN advantages/drawbacks

The advantages of the KNN method are that it is a simple and interpretable model. Since we do not have many data, the computation cost will be low. Thus, it is the first model that we test. Nevertheless, as this method is sensitive to large dimensions, it might not have the highest performance on our dataset seeing there are 50 columns after encoding and only 383 observations.



Confusion matrix

ROC curve

Figure 9: Confusion matrix and ROC curve obtained with the KNN method

4.2 Model 2: [Logistic Regression (LR)]

During model training, we use 5-fold cross-validation to select the best parameter (the parameter of regularisation) from a well-chosen grid of candidate values.

We use the logistic regression model with lasso regularisation to obtain the following classifier:

$$\begin{aligned}\boldsymbol{P(Y=1\mid x)\approx f(x)=\sigma(}&\mathbf{0.356}\cdot\text{Age}-\mathbf{0.104}\cdot\text{Focality}+\mathbf{0.111}\cdot\text{Physical Examination\_Multinodular goiter}\\&+\mathbf{0.111}\cdot\text{Pathology\_Papillary}+\mathbf{0.079}\cdot\text{Risk\_High}-\mathbf{1.166}\cdot\text{Risk\_Low}-\mathbf{0.022}\cdot\text{T\_T2}\\&-\mathbf{0.984}\cdot\text{N\_N0}+\mathbf{0.414}\cdot\text{N\_N1a}-\mathbf{0.303}\cdot\text{Stage\_I}\\&+\mathbf{0.646}\cdot\text{Response\_Biochemical Incomplete}\\&-\mathbf{2.739}\cdot\text{Response\_Excellent}-\mathbf{0.738}\cdot\text{Response\_Indeterminate}\\&+\mathbf{4.127}\cdot\text{Response\_Structural Incomplete})\end{aligned}$$

where $\boldsymbol{\sigma(z)}$ denotes the sigmoid function:

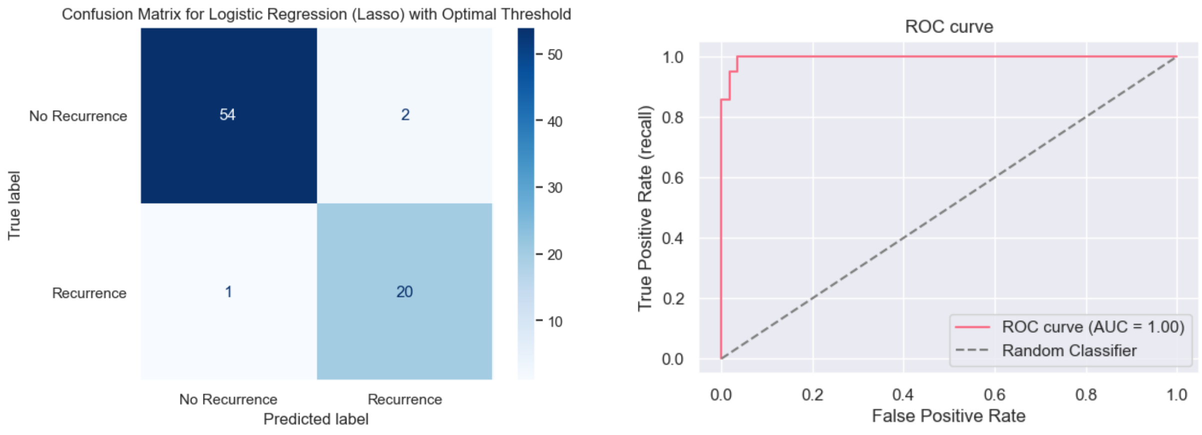
$$\boldsymbol{\sigma(z) = \frac{1}{1+e^{-z}}}$$

This model will predict that a patient $\boldsymbol{x}$ will be classified as a recurrence case if $\boldsymbol{f(x) \geq 0.211}$.

4.2.1 LR advantages/drawbacks

In one hand, the drawbacks are that the method does not capture non-linear interactions, so it is limited, and it assumes that the classes are linearly separable. On the other hand, the advantages are that the method is fast to train and, with L1 regularisation, it is more robust to overfitting. In particular, a key benefit of this model is its interpretability: we use L1 regularisation to encourage sparsity, which leads to simpler and more readable models.

Interpretation. The final logistic regression model enables us to interpret the effect of each selected feature on the probability of cancer recurrence. Features not appearing in the equation are assumed to have no significant effect in the final model.
Among all variables, the most decisive factor is the patient's treatment response. The coefficient for `Response_Excellent` is strongly negative (-2.739), indicating a strong protective effect. Similarly, `Response_Indeterminate` also lowers the risk of recurrence (-0.738), although to a minor extent. In contrast, `Response_Structural Incomplete` is associated with a much higher risk of recurrence, as reflected by the large positive coefficient (+4.127). Finally, `Response_Biochemical Incomplete` also has a positive effect (+**0.646**), though it is more moderate.
The variable `Risk_Low` is another strong protective factor with a coefficient of -1.166, while `Risk_High` has a small positive effect (+0.079).
Regarding cancer staging variables, `N_N0` (-0.984) and `Stage_I` (-0.303) both reduce the probability of recurrence. Conversely, `N_N1a` (+0.414) increases it.
Other variables such as `Age` (+0.356), `Pathology_Papillary` (+0.111), `Focality` (-0.104), and `Physical Examination_Multinodular goiter` (+0.111) have more moderate but non-negligible contributions.



Confusion matrix

ROC curve

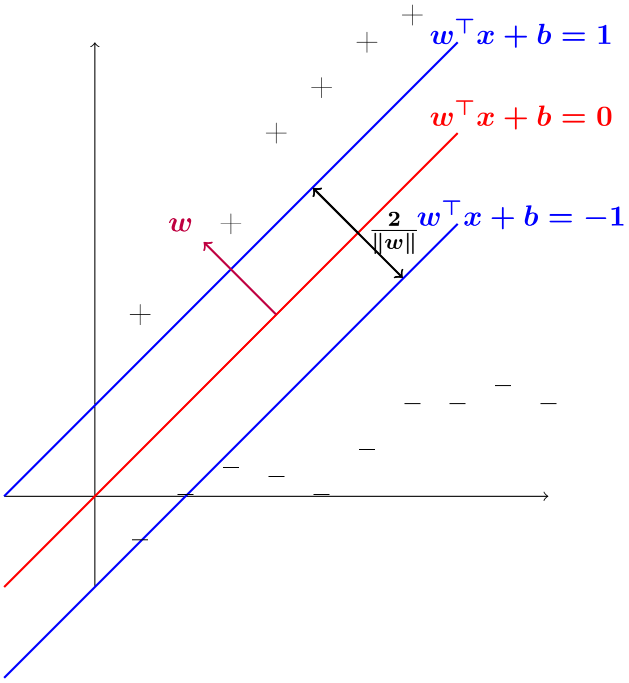
Figure 10: Confusion matrix and ROC curve obtained with the Logistic regression method

4.3 Model 3: [Support Vector Machine (SVM)]

4.3.1 Explanation: Linear Case

Logistic regression separates the data using a hyperplane, and the results are promising. Support Vector Machines (SVM) aim to find the best separating hyperplane by maximizing the margin. This encourages us to test this alternative method.
We assign the label $+1$ to the class "yes" and -1 to the class "no".

Geometric Interpretation of SVM



The SVM seeks a linear hyperplane $w^\top x + b = 0$ such that:

$$\forall i, \quad \begin{cases} w^\top x_i + b \geq +1 & \text{if } y_i = +1 \\ w^\top x_i + b \leq -1 & \text{if } y_i = -1 \end{cases}$$

This can be written more compactly as a single inequality:

$$y_i(w^\top x_i + b) \geq 1 \quad \forall i$$

The margin between the two classes is given by $\frac{2}{\|w\|}$, so maximizing the margin corresponds to minimizing $\|w\|^2$. The resulting optimization problem is:

$$\begin{aligned} \min_{w,\, b} \quad & \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i(w^\top x_i + b) \geq 1 \quad \forall i \end{aligned}$$

This model is called a *hard-margin SVM*. It assumes that the data are linearly separable and it is sensitive to outliers. To overcome this limitation, we use the *soft-margin SVM*:

$$\boxed{\begin{aligned} \min_{w,\, b,\, \xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i \end{aligned}}$$

The larger the parameter C, the less the model tolerates margin violations.

4.3.2 Explanation: Nonlinear Case

When the classes are not linearly separable, we apply the *kernel trick*. The idea is to transform the data into another feature space $\boldsymbol{\mathcal{X}}$ where they become linearly separable.
The SVM formulation only involves inner products of the form:

$$\boldsymbol{x}_i^\top \boldsymbol{x}_j.$$

If we apply a nonlinear transformation:

$$\boldsymbol{\phi} : \mathbb{R}^n \to \boldsymbol{\mathcal{X}}, \quad \boldsymbol{x} \mapsto \boldsymbol{\phi}(\boldsymbol{x}),$$

we need to compute:

$$\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j).$$

The kernel trick avoids computing $\boldsymbol{\phi}$ explicitly by introducing a kernel function:

$$\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j).$$

In our project, we focus on two well-known kernels:

- **Radial Basis Function (RBF) kernel:**

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)$$

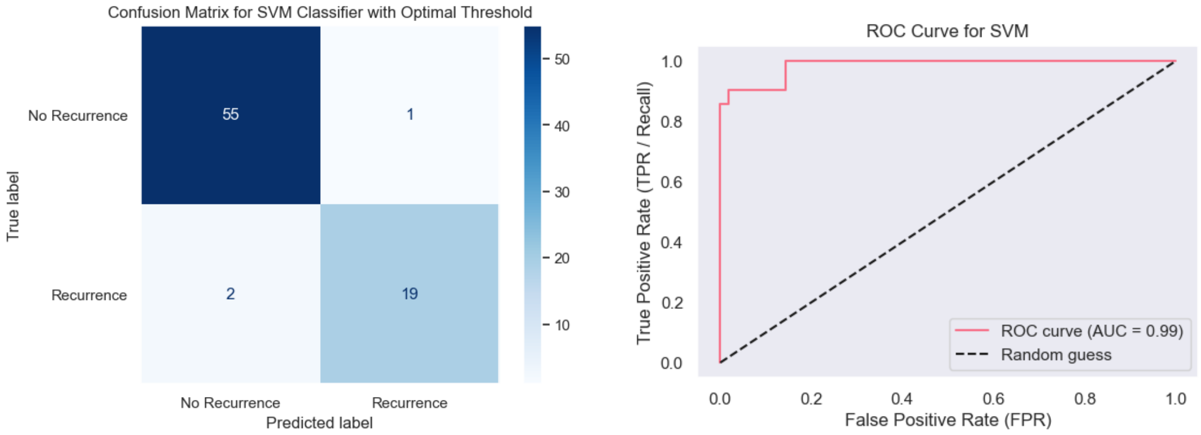
- **Polynomial kernel:**

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') = (\gamma\, \boldsymbol{x}^\top \boldsymbol{x}' + \boldsymbol{r})^{\boldsymbol{d}}$$

During model training, we use 5-fold cross validation to select the best parameters (we define the kernel, the parameters C and γ, the degree of the polynomial kernel; and we choose if the weights of the classes are balanced or not) from a well-chosen grid of candidate values.

4.3.3 SVM advantages/drawbacks

The advantages of SVM are that it works even when the classes are not linearly separable, and it gives accurate results when the number of observations is not very large (which is the case here). However, the model is not interpretable, it is slower to train than logistic regression, and it uses an approximation of probabilities.



Confusion matrix

ROC curve

Figure 11: Confusion matrix and ROC curve obtained with the SVM

4.4 Model 4: [Random Forest (RF)]

4.4.1 Explanation of Random Forest

Random Forest Models are a completely different approach to the previous ones. They are based on the idea of ensemble learning, which consists of combining multiple models to improve performance. In this case, we will use a collection of random decision trees, each trained on a random subset of the data and features. These models consists as the construction of multiple decision trees, and each of them predicts the class of a given sample. The final prediction is made by aggregating the predictions of all trees, typically by majority voting for classification tasks. If we wanted to use a single tree to do the prediction, we would need a tree with high depth but this would lead to a small bias and a high variance, that is overfitting. Using a collection of random decision trees is a way of reducing the variance. For sure this leads to a small increase of the bias. A decision tree partitions the input space into regions by recursively splitting the data based on feature values. At each node, it selects a feature and a rule that best separates the data into distinct groups. Classification is made by following the path down the tree to a leaf, which assigns the predicted class. So training a decision tree consists of finding the best features and rules to split the data at each node, which is done by minimizing a loss function (e.g., Gini impurity or entropy for classification tasks). The main point from the random decision tree to the random forest is how we construct and train distinct trees. The first idea is the bagging (Bootstrap Aggregating) method, which consists of training each tree on a random subset of the data (select random sample with replacement), this method is useful to decrease the correlation between the trees. The second idea is to use a random subset of features at each split in the tree, which also helps to remove the correlation among the trees and further reduce variance.

4.4.2 Pre-processing for Random Forest

For all models using trees the One-Hot-Encoder method is not necessary so we will not use it. The categorical variables will be encoded as integers, which is sufficient for the decision tree models. The numeric variable `Age` will be kept as is, without any transformation.

4.4.3 Choice of Hyperparameters

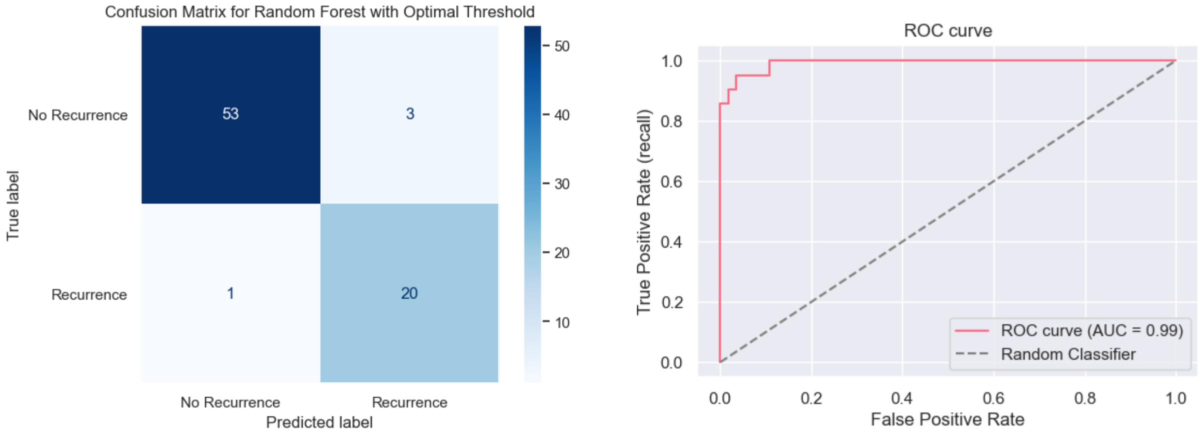
The main hyperparameters to tune in a Random Forest model are:

- **Number of trees (n_estimators):** The number of decision trees in the forest. More trees generally lead to better performance but increase computation time.
- **Maximum depth (max_depth):** The maximum depth of each tree. Limiting the depth can help prevent overfitting.
- **Minimum samples per leaf (min_samples_leaf):** The minimum number of samples required to be at a leaf node. This can help control overfitting.
- **Minimum samples to split (min_samples_split):** The minimum number of samples required to split an internal node. This can also help control overfitting.

To estimate the best hyperparameters, we will use a grid search with 5 folds cross-validation and using the Recall metric as explained before. We also compute the adequate class weight to balance the classes, as the dataset is unbalanced.
Finally, the optimization of the threshold probability for classification in this model, helps us to have a robust model with small number of false negative prediction.

4.4.4 Advantages/Drawbacks of Random Forest

The main advantages of Random Forest models are the quality of predictions and the computational efficiency. Nonetheless, there is a lack of explainability, as it is difficult to interpret the results of a Random Forest model. Indeed, the model is a collection of many random decision trees, and it is not easy to understand how the model makes its predictions.



Confusion matrix

ROC curve

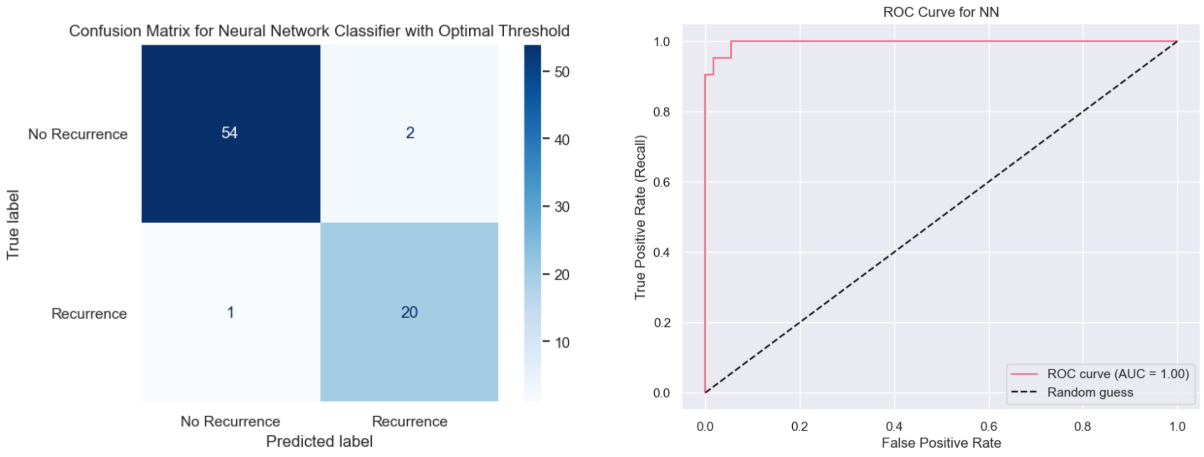
Figure 12: Confusion matrix and ROC curve obtained with the Random Forest model

4.5 Model 5: [Dense Neural Network (DNN)]

For this method we split the validation into two sets (50%,50%). The first subset is used to find the best architecture and the second subset is used to optimize the threshold. On each hidden layer, there is a ReLU activation, and the final activation is a sigmoid in order to get an output between 0 and 1 alike a probability.

4.5.1 DNN advantages/drawbacks

This method is able to see more complex relationships than the linear ones and directly returns a probability in our case (with sigmoid as last activation function) so it is easy to optimize the threshold (in the sense that we do not need to approximate the probabilities). Nevertheless, the model is not interpretable and can be time-consuming because it need to test many combinations of architectures.



Confusion matrix

ROC curve

Figure 13: Confusion matrix and ROC curve obtained with the Dense neural network

5 Performance Comparisons

Table 5 below summarizes the performance metrics of the different models, allowing for a direct comparison of their Accuracy, Recall and AUC obtained on the test set .

Table 5: Comparison of model performance based on accuracy, recall, and AUC on the test set

Model	**Accuracy**	**Recall**	**AUC**
K-Nearest Neighbors	0.92	0.81	0.96
Logistic Regression	0.96	0.95	$\approx$ 1.00
SVM	0.96	0.90	0.99
Random Forest	0.95	0.95	0.99
Dense Neural Network	0.96	0.95	$\approx$ 1.00

6 Conclusion

In this study, we evaluated several supervised learning models to predict the recurrence of thyroid cancer based on clinical and pathological features. Our analysis focused on three key performance metrics: accuracy, recall, and AUC.

Compared to the other models, the Dense Neural Network and Logistic Regression achieved the best results. Random Forest and SVM performed well, offering high AUC and competitive recall values. However, the K Nearest Neighbors algorithm showed the weakest recall (0.81), which may limit its clinical applicability, especially given the importance of minimizing false negatives in this context.

Although both Dense Neural Network and Logistic Regression model yield similar performance in terms of accuracy, recall, and AUC, the DNN has the potential to capture non-linear interactions between covariates. This flexibility may become advantageous in more complex datasets or when subtle feature interactions are present.

Nonetheless, given their identical results in our specific case, Logistic Regression may be preferred due to its simplicity, transparency, interpretability and qualities that are particularly valuable in a clinical context. It allows specialist to understand and trust the model's decisions more easily, which is essential when guiding medical decisions.

One limitation shared by all introduced models is that the classification threshold was optimized using a single validation set, with a relatively small size (around 70 observations, or approximately 35 in the case of the DNN). As a result, the robustness of the chosen threshold may be suboptimal. To address this, one could apply cross validation for threshold optimization, which would provide more stable and generalizable results. Nevertheless, this approach would significantly increase computational costs.

Appendix

This section can be considered outside the main scope of the project as the results were not conclusive. Yet, it represents an attempt to build a more robust Neural Network using standard 5-fold cross validation to identify the best architecture. Following this, we applied an alternative approach, the aggregated cross validation (5-fold) to further evaluate the model's performance and limit the risk of overfitting.

What is the aggregated cross validation ?

This is a cross validation approach where we use CV to collect out-of-fold predictions, then optimize the threshold on these aggregated predictions. This method ensures that the threshold is robust and generalizable.

Step 1: Cross-Validation Setup

- Creates 5 stratified folds (maintains class distribution in each fold);
- Each fold will serve as validation data once.

```
kfold_thresh = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
```

Step 2: Collect Out-of-Fold Predictions

```
for fold, (train_idx, val_idx) in enumerate(kfold_thresh.split(X, y)):
    # Train model on 4/5 of the data
    model.fit(X_fold_train, y_fold_train, ...)

    # Predict on the remaining 1/5 (validation fold)
    y_proba_fold = model.predict(X_fold_val, verbose=0)

    # Collect ALL validation predictions and true labels
    all_y_true.extend(y_fold_val)
    all_y_proba.extend(y_proba_fold.flatten())
```

Each prediction in `all_y_proba` comes from a model that never saw that specific data point during training.

Step 3: Aggregate Results

- `all_y_true`: Contains true labels for all training data points;
- `all_y_proba`: Contains out-of-fold predictions for all training data points;
- Each prediction is "out-of-sample" relative to the model that made it.

Step 4: Single Threshold Optimization

```
precisions, recalls, thresholds = precision_recall_curve(all_y_true, all_y_proba)
```

- Uses the predictions to compute the precision-recall curve;
- Finds optimal threshold using F$\boldsymbol{\beta}$-score on these out-of-fold predictions.

In the classical method, we would have computed an optimal threshold for each fold and considered, for example, the mean.

Advantages of This Approach

- **No Data Leakage:** The threshold is optimized on predictions that are truly "unseen" by their respective models;
- **Robust Estimation:** Uses all available training data for threshold selection;
- **Better Generalization:** The optimized threshold should work well on new data;
- **Efficient:** Only requires one precision-recall curve computation (not 5 separate ones).

Comparison with Alternative Approaches

Table 6: Comparison of threshold selection methods

Method	**Data Leakage Risk**	**Threshold Robustness**	**Computational Cost**
Single Train/Val Split	Low	Low (single split)	Low
CV Aggregation (Our Method)	None	High (all data used)	Medium
Separate CV per Threshold	None	Medium	Very High

This method strikes the optimal balance between robustness, computational efficiency, and methodological rigor, though it is also time-consuming.

References

[1] Shiva Borzooei and Aidin Tarokhian. *Differentiated Thyroid Cancer Recurrence*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5632J. 2023.