

Introduction to classification model

Tengku Muhammad Hanis Bin Tengku Mokhtar, PhD

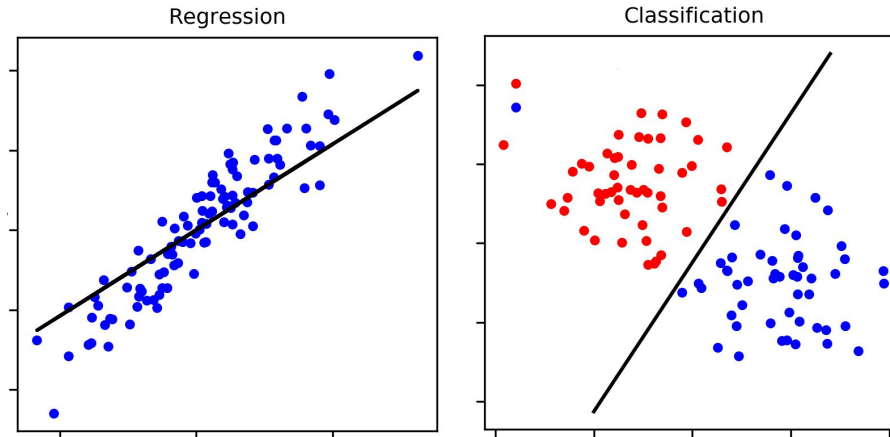
October 23, 2024

Contents

- [Classification](#)
- [Decision tree](#)
- [Random forest](#)
- [Artificial neural network \(ANN\)](#)
- [Performance metrics](#)
- [Multiclass performance metrics](#)
- [Suggested readings](#)

Classification

- A supervised ML task in which the model predict the categorical outcome
- Classification task can be further divided into:
 - Binary or two class classification
 - Multiclass classification

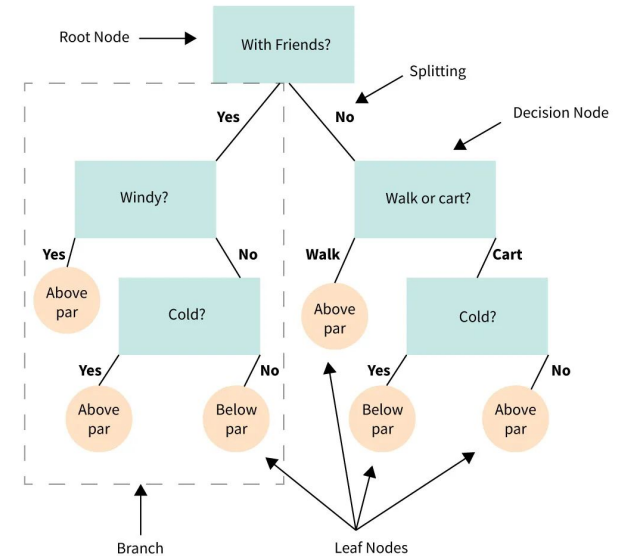


Classification algorithms

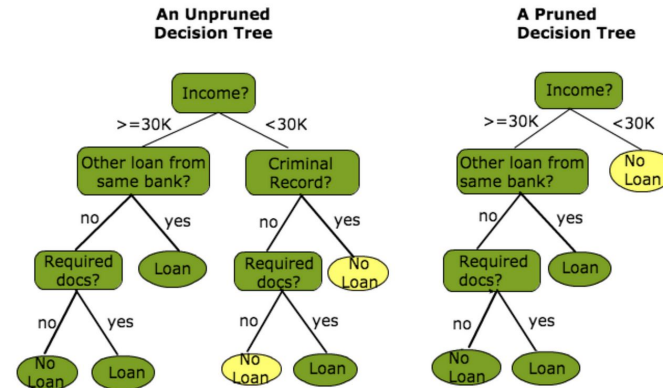
- Just to list a few:
 - Logistic Regression
 - Decision Trees
 - Random Forest
 - Support Vector Machines (SVM)
 - k-Nearest Neighbors (kNN)
 - Naive Bayes
 - Artificial Neural Networks (ANN)
- [Full list of algorithms in parsnip package](#)

Decision tree

- Can be used for regression and classification - classification and regression trees (CART) models
- The order of the variable to be splitted is determined by the purity:
 - Gini impurity
 - Entropy and information gain
- Purity - how well separated the points are at the nodes

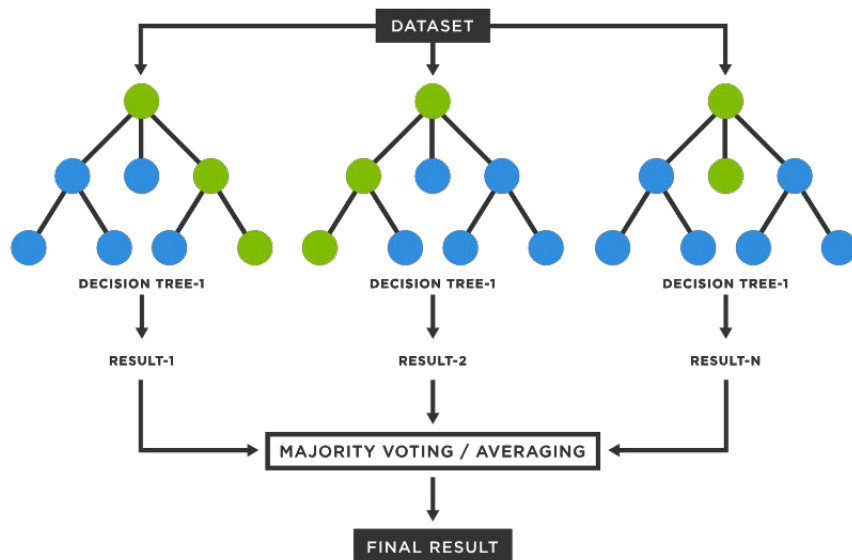


- Pros:
 - Easy to understand
 - Fast computation
 - Able to handle missing data and outliers
- Cons:
 - Tree overfitting - can be overcome with a pruning or CV methods



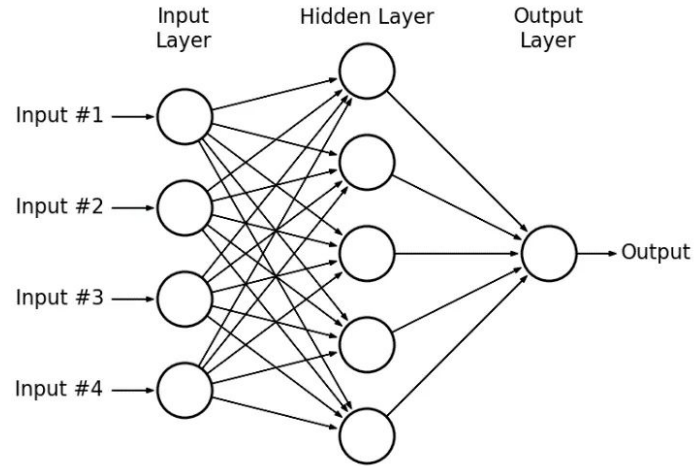
Random forest

- Basically, a collection of decision tree
- Pros:
 - Low risk of overfitting
 - Usually more accurate
 - Able to handle missing data and outliers
- Cons:
 - Relatively slow computation
 - Low interpretability

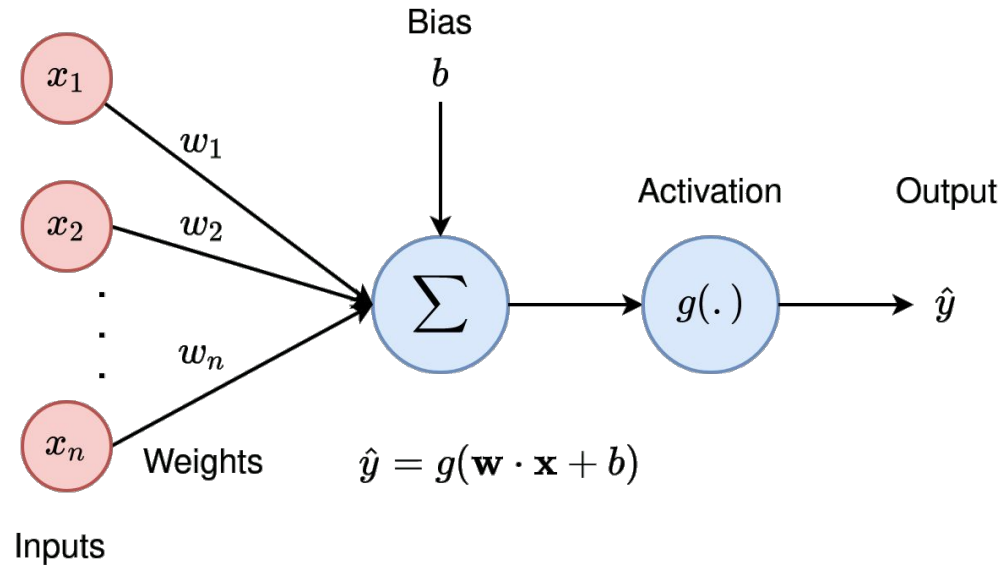


Artificial Neural Network (ANN)

- Neural network is the basis of deep learning
- ANN can be used for regression and classification
- Simple ANN is a single layer, feed-forward neural network, which also known as multilayer perceptron (MLP)
- ANN is formed of:
 - Input layer
 - Hidden layer
 - Output layer



- Breakdown of ANN:



Performance metrics

Confusion matrix

- It is a table comparing the predicted and the actual classes
- Best confusion matrix is at a [wiki page](#)

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Accuracy

- Proportion of correctly classified cases out of the total cases

Sensitivity/recall

- High sensitivity means effective at detecting the true positive cases
- High sensitivity means low false negative

Specificity

- High specificity means effective at detecting the true negative cases
- High specificity means low false positive

Precision/positive predictive value (PPV)

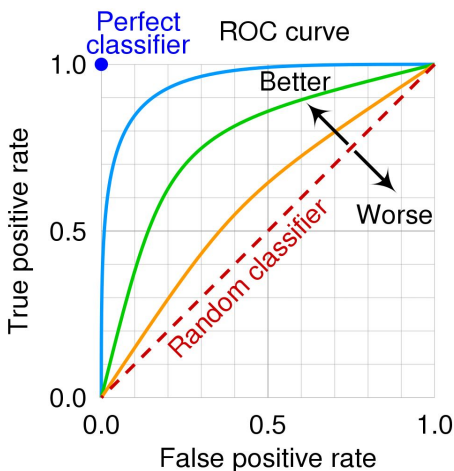
- Indicates proportion of subjects with a predicted positive who truly positive
- High precision - low false positive

Negative predictive value (NPV)

- Indicates proportion of subjects with a predicted negative who truly negative
- High NPV - low false negative

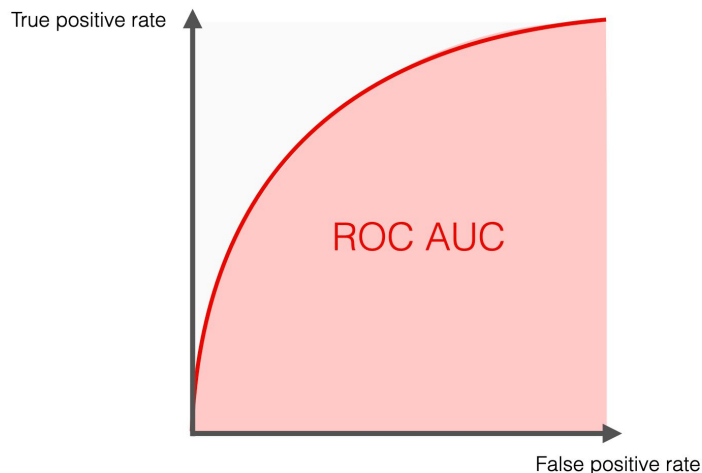
Receiver operating characteristic (ROC) curve

- Reflects a performance of classification models at certain threshold (usually 0.5)
- Can be used to compare different classification ML models



ROC-Area under the curve (ROC-AUC)

- It provides an aggregate measure of the model's performance
- AUC of 0.5 = no discriminant ability, while AUC of 1 = perfect classification
- Can be used to compare different classification ML models



Multiclass performance metrics

- There are at least 3 methods:
 - Macro averaging
 - Weighted macro averaging
 - Micro averaging

The diagram illustrates the relationship between the general precision formula and its three multiclass variants. A central box on the left contains the formula for precision: $Pr = \frac{TP}{TP + FP}$. Three arrows originate from the right side of this box and point to three separate boxes on the right, each containing a formula for a different multiclass performance metric.

General Precision Formula:

$$Pr = \frac{TP}{TP + FP}$$

Macro Averaging:

$$Pr_{macro} = \frac{Pr_1 + Pr_2 + \dots + Pr_k}{k} = Pr_1 \frac{1}{k} + Pr_2 \frac{1}{k} + \dots + Pr_k \frac{1}{k}$$

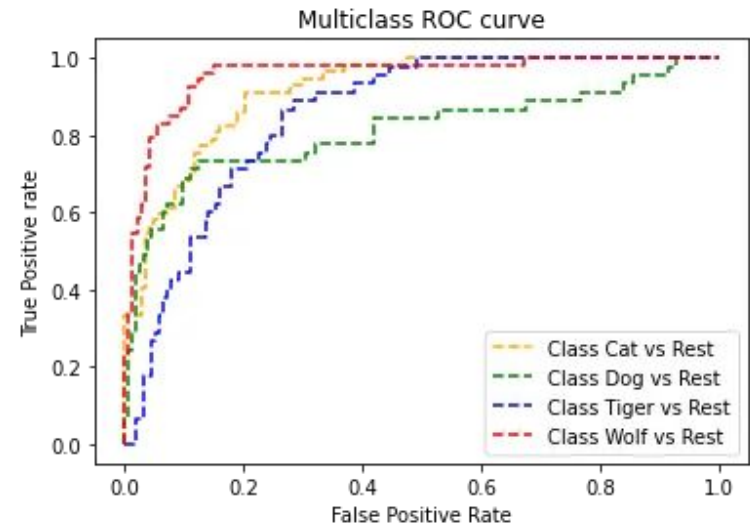
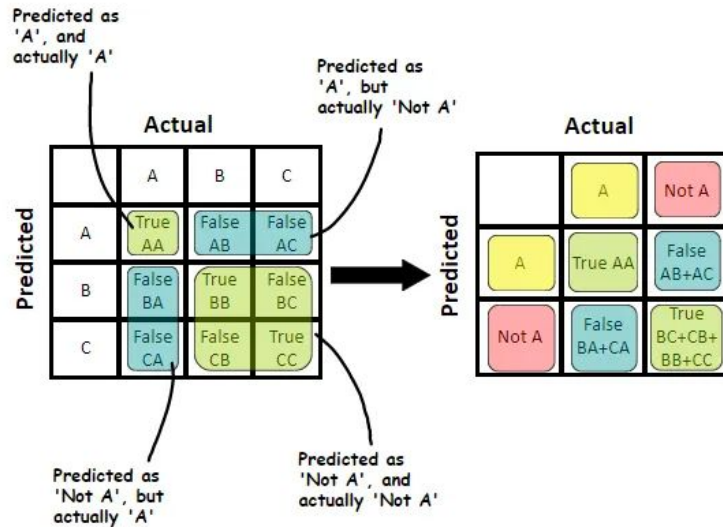
Weighted Macro Averaging:

$$Pr_{weighted-macro} = Pr_1 \frac{\#Obs_1}{N} + Pr_2 \frac{\#Obs_2}{N} + \dots + Pr_k \frac{\#Obs_k}{N}$$

Micro Averaging:

$$Pr_{micro} = \frac{TP_1 + TP_2 + \dots + TP_k}{(TP_1 + TP_2 + \dots + TP_k) + (FP_1 + FP_2 + \dots + FP_k)}$$

- For metrics such as ROC and ROC-AUC, one versus all method can be used



Suggested readings/references

- Kuhn, M., & Silge, J. (2022). [Tidy Modeling with R: A Framework for Modeling in the Tidyverse](#). O'Reilly Media.
- Burger, S. V. (2018). Introduction to machine learning with R: Rigorous mathematical analysis (First edition). O'Reilly Media.



Any question?



tengkuhanismokhtar@gmail.com
jom.research.malaysia@gmail.com