





Deep learning for agronomy


3rd year engineer

Ammouri Bilel

 : ESAM

 : ammouri-bilel

 : bilelammouri

 : Ammouri-Bilel

 : 0000-0002-5491-5172

Plan

- 1 Classification models in ML
- 2 Metrics to Evaluate ML Classification Algorithms

Classification analysis

Classification is a form of supervised machine learning in which the model endeavors to predict the accurate label of input data. During the training phase, the model is comprehensively trained using a labeled dataset, and subsequently, its performance is assessed using a separate test dataset before deploying it for predictions on novel, unseen data.

Types of Classification models

There are four primary classification tasks in machine learning:

- ① binary classifications
- ② multi-class classifications
- ③ multi-label classifications
- ④ imbalanced classifications.

Types of Classification models

Binary classifications

In binary classification, the objective is to categorize input data into two mutually exclusive categories. The training data is typically labeled in a binary format, such as true and false, positive and negative, 0 and 1, or spam and not spam, depending on the specific problem at hand.

For example, one might seek to determine whether a given image depicts a strawberry or a cherry.

Algorithms like Logistic Regression and Support Vector Machines are inherently designed for binary classifications.

However, other algorithms like K-Nearest Neighbors and Decision Trees can also be applied to binary classification tasks.

Types of Classification models

Multi-class classifications

Multi-class classification involves assigning input data to at least two mutually exclusive class labels, with the objective of predicting the specific class to which a given input example belongs.

While many algorithms designed for binary classification can also be utilized for multi-class tasks, including but not limited to:

- Logistic Regression
- K-Nearest Neighbors
- Naive Bayes
- SVM
- Random Forest

It's important to note that SVM and Logistic Regression don't inherently support multi-class classification. However, we can overcome this limitation by employing binary transformation approaches, such as one-versus-one and one-versus-all. This allows us to adapt native binary classification algorithms to effectively handle multi-class classification tasks.

Types of Classification models

Multi-label classifications

In the realm of multi-label classification tasks, the objective is to predict zero or more classes for each input example. Notably, there is no mutual exclusion, as a single input example can be associated with multiple labels.

This dynamic is evident in various domains, such as auto-tagging in Natural Language Processing, where a given text may encompass multiple topics. Similarly, in computer vision, an image can be adorned with various objects, as depicted below: the model correctly predicted that the image encompasses a tree, an insect, a leaf, and a strawberry.

It's crucial to recognize that employing multi-class or binary classification models is impractical for multi-label classification due to the non-exclusive nature of the labels.

Many algorithms tailored for multi-label classification :

- Multi-label Decision Trees
- Multi-label Gradient Boosting
- Multi-label Random Forests

Types of Classification models

Imbalanced classifications

In imbalanced classification scenarios, the distribution of examples across classes is uneven, resulting in a disparity where one class may have a significantly higher representation than others in the training data. Consider a 3-class classification scenario with training data consisting of 70% trees, 25% strawberries, and 5% insects.

Imbalanced classification challenges are prevalent in various real-world situations, including:

- Fraudulent transaction detection in financial industries
- Rare disease diagnosis
- Customer churn analysis

Types of Classification models

Imbalanced classifications

Traditional predictive models like Decision Trees and Logistic Regression may prove ineffective when confronted with imbalanced datasets. They risk biasing predictions toward the class with the highest number of observations, potentially dismissing those with fewer instances as mere noise.

However, the story doesn't end there. Several approaches can be employed to address imbalanced datasets. Common strategies involve the use of **sampling techniques** or **leveraging the capabilities of cost-sensitive algorithms**. These techniques play a crucial role in ensuring that the model doesn't disproportionately favor the majority class, thereby enhancing its ability to discern patterns in the minority classes.

Types of Classification models

Imbalanced classifications

Sampling Techniques

Sampling techniques play a pivotal role in addressing the imbalance in class distribution within a dataset. These methods strive to bring about a more equitable representation by employing strategies such as:

- Cluster-based Oversampling
- Random undersampling (majority class)
- SMOTE Oversampling: (minority class)

Cost-Sensitive Algorithms

Cost-sensitive algorithms take into account the potential cost associated with misclassification. Their objective is to minimize the overall cost incurred by the model. Notable examples include:

- Cost-sensitive Decision Trees
- Cost-sensitive Logistic Regression
- Cost-sensitive Support Vector Machines

Performance measures

Definition

Performance measures for classification models are metrics used to evaluate how well a model is performing in predicting class labels. They provide insights into different aspects of the model's behavior, such as accuracy, precision, recall, and the ability to discriminate between positive and negative instances.

Performance measures

Confusion Matrix

A confusion matrix is a table that summarizes the performance of a classification algorithm. It breaks down the predictions into four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Performance measures

Confusion Matrix

		Condition Phase (worst case)		Actual
		Condition Positive/ Shaded	Condition Negative/ Unshaded	
Testing Phase (best case)	Test Positive/ Shaded	True positive shaded T_p (Correct)	False positive shaded F_p (Incorrect)	Precision/Positive Predictive Value (PPV) $\frac{T_p}{T_p + F_p} \times 100\%$
	Test Negative/ Unshaded	False negative unshaded F_n (Incorrect)	True negative unshaded T_n (Correct)	Negative Predictive Value (NPV) $\frac{T_n}{T_n + F_n} \times 100\%$
		Sensitivity/Recall Rate (RR) $\frac{T_p}{T_p + F_n} \times 100\%$	Specificity Rate (SR) $\frac{T_n}{T_n + F_p} \times 100\%$	

Performance measures

ROC Curve (Receiver Operating Characteristic)

The ROC curve is a graphical representation of a model's performance across different threshold settings. It plots the true positive rate (sensitivity) against the false positive rate ($1 - \text{specificity}$) at various threshold values.

Formula

ROC Curve is created by plotting TPR (True Positive Rate) against FPR (False Positive Rate) at different threshold values.

- A value below 0.5 indicates a poor classifier
- A value of 0.5 means a random classifier
- Value over 0.7 corresponds to a good classifier
- 0.8 indicates a strong classifier

Strategies to choose the right metric

1 Choose accuracy:

- The cost of F_p and F_n are roughly equal
- The benefit of T_p and T_n are roughly equal

2 Choose precision:

- The cost of F_p is much higher than a F_n
- The benefit of a T_p is much higher than a T_n

3 Choose recall:

- The cost of F_n is much hogher than a F_p
- The cost of a T_n is much higher than a T_p

4 ROC/AUC:

- Use ROC when the dealing with balanced data sets
- Use precision-recall for imbalanced data sets