

# Package ‘TidyML’

May 21, 2025

**Title** Machine Learning Modelling For Everyone

**Version** 0.0.0.9000

## Description

TidyML is a minimal library focused on providing all the essential tools for the workflow of a machine learning modelling process. The whole process is divided into 5 steps:  
preprocessing() -> build\_model() -> fine\_tuning() -> show\_results() -> sensitivity\_analysis()

**License** `use\_mit\_license()`, `use\_gpl3\_license()` or friends to pick a license

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Depends** R (>= 2.10)

**Imports** R6,

tidyr,  
tidyselect,  
magrittr,  
dials,  
parsnip,  
recipes,  
rsample,  
tune,  
workflows,  
yardstick,  
vip,  
glue,  
ggpubr,  
innsight,  
torch,  
shapr,  
DiagrammeR,  
ggbeeswarm

**Suggests** testthat (>= 3.0.0),

brulee,  
ranger,  
kernlab,  
xgboost,  
dplyr,

ggplot2,  
rlang,  
tibble,  
tidyverse

**Config/testthat/edition** 3

**URL** <https://github.com/JMartinezGarcia/TidyML>

**BugReports** <https://github.com/JMartinezGarcia/TidyML/issues>

**LazyData** true

Contents

build_model . . . . .	2
fine_tuning . . . . .	4
preprocessing . . . . .	5
sensitivity_analysis . . . . .	6
show_results . . . . .	6
sim_data . . . . .	8
<b>Index</b>	<b>9</b>

---

build_model	<i>Create ML Model</i>
-------------	------------------------

---

Description

Create ML Model

Usage

build\_model(analysis\_object, model\_name, hyperparameters = NULL)

Arguments

- analysis\_object  
analysis\_object created from preprocessing function.
- model\_name  
Name of the ML Model. A string of the model name: "Neural Network", "Random Forest", "SVM" or "XGBOOST".
- hyperparameters  
Hyperparameters of the ML model. List containing the name of the hyperparameter and its value or range of values.

Value

Updated analysis\_object

## Hyperparameters

### Neural Network:

Parsnip model using **brulee** engine. Hyperparameters:

- **hidden\_units**: Number of Hidden Neurons. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(5, 20)`.
- **activation**: Activation Function. A vector with any of ("relu", "sigmoid", "tanh") or NULL for default values `c("relu", "sigmoid", "tanh")`.
- **learn\_rate**: Learning Rate. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(-3, -1)` in log10 scale.

### Random Forest:

Parsnip model using **ranger** engine. Hyperparameters:

- **trees**: Number of Trees. A single value, a vector with range values `c(min_val, max_val)`. Default range `c(100, 300)`.
- **mtry**: Number of variables randomly selected as candidates at each split. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(3, 8)`.
- **min\_n**: Minimum Number of samples to split at each node. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(2, 25)`.

### XGBOOST:

Parsnip model using **xgboost** engine. Hyperparameters:

- **trees**: Number of Trees. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(100, 300)`.
- **mtry**: Number of variables randomly selected as candidates at each split. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(3, 8)`.
- **min\_n**: Minimum Number of samples to split at each node. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(5, 25)`.
- **tree\_depth**: Maximum tree depth. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(3, 10)`.
- **learn\_rate**: Learning Rate. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(-4, -1)` in log10 scale.
- **loss\_reduction**: Minimum loss reduction required to make a further partition on a leaf node. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(-5, 1.5)` in log10 scale.

### SVM:

Parsnip model using **kernlab** engine. Hyperparameters:

- **cost**: Penalty parameter that regulates model complexity and misclassification tolerance. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(-3, 3)` in log10 scale.
- **margin**: Distance between the separating hyperplane and the nearest data points. A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(0, 0.2)`.
- **type**: Kernel to be used. A single value from ("linear", "rbf", "polynomial")
- **rbf\_sigma**: A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(-5, 0)` in log10 scale.
- **degree**: Polynomial Degree (polynomial kernel only). A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(1, 3)`.
- **scale\_factor**: Scaling coefficient applied to inputs. (polynomial kernel only) A single value, a vector with range values `c(min_val, max_val)` or NULL for default range `c(-5, -1)` in log10 scale.

---

fine_tuning	<i>Fine Tune ML Model</i>
-------------	---------------------------

---

## Description

Fine Tune ML Model

## Usage

```
fine_tuning(analysis_object, tuner, metrics, plot_results = F, verbose = FALSE)
```

## Arguments

analysis_object	analysis_object created from build_model function.
tuner	Name of the Hyperparameter Tuner. A string of the tuner name: "Bayesian Optimization" or "Grid Search CV".
metrics	Metric used for Model Selection. A string of the name of metric (see Metrics).
plot_results	Whether to plot the tuning results. Boolean TRUE or FALSE (default).
verbose	Whether to show tuning process. Boolean TRUE or FALSE (default).

## Value

Updated analysis\_object

## Tuners

### Bayesian Optimization:

- Initial data points: 20
- Maximum number of iterations: 25
- Convergence after 5 iterations without improvement
- Train / Validation / Test : 0.6 / 0.2 / 0.2

### Grid Search CV:

- Number of Folds: 5
- Maximum levels per hyperparameter: 10
- Train / Test : 0.75 / 0.25

## Metrics

### Regression Metrics:

- rmse
- mae
- mpe
- mape
- ccc
- smape
- rpiq

- rsq

#### Classification Metrics:

- accuracy
- bal\_accuracy
- recall
- sensitivity
- specificity
- kap
- f\_meas
- mcc
- j\_index
- detection\_prevelance
- roc\_auc
- pr\_auc
- gain\_capture
- brier\_class
- roc\_aunp

preprocessing

*Preprocessing Data Matrix*

### Description

Preprocessing Data Matrix

### Usage

```
preprocessing(
  df,
  formula,
  task = "regression",
  num_vars = NULL,
  cat_vars = NULL,
  norm_num_vars = "all",
  encode_cat_vars = "all",
  y_levels = NULL
)
```

### Arguments

df	Input DataFrame. Either a data.frame or tibble.
formula	Modelling Formula. A string of characters or formula.
task	Modelling Task. Either "regression" or "classification".
num_vars	Optional vector of names of the numerical features.
cat_vars	Optional vector of names of the categorical features.
norm_num_vars	Normalize numeric features as z-scores. Either vector of names of numerical features to be normalized or "all" (default).

encode\_cat\_vars

One Hot Encode Categorical Features. Either vector of names of categorical features to be encoded or "all" (default).

y\_levels

Optional ordered vector with names of the target variable levels (Classification task only).

### Value

An analysis\_object

---

sensitivity\_analysis    *Perform Sensitivity Analysis and Interpretable ML methods*

---

### Description

Perform Sensitivity Analysis and Interpretable ML methods

### Usage

```
sensitivity_analysis(analysis_object, methods = c("PFI"), metric = NULL)
```

### Arguments

analysis\_object

analysis\_object created from fine\_tuning function.

methods

Method to be used. A string of the method name: "PFI" (Permutation Feature Importance), "SHAP" (SHapley Additive exPlanations), "Integrated Gradients" (Neural Network only) or "Olden" (Neural Network only).

metric

Metric used for "PFI" method (Permutation Feature Importance). A string of the name of metric (see Metrics).

### Value

Updated analysis\_object

---

show\_results    *Showcase Summary Results and Plots*

---

### Description

Showcase Summary Results and Plots

**Usage**

```
show_results(
  analysis_object,
  summary = FALSE,
  roc_curve = FALSE,
  pr_curve = FALSE,
  gain_curve = FALSE,
  lift_curve = FALSE,
  dist_by_class = FALSE,
  reliability_plot = FALSE,
  confusion_matrix = FALSE,
  scatter_residuals = FALSE,
  scatter_predictions = FALSE,
  residuals_dist = FALSE,
  new_data = "test"
)
```

**Arguments**

analysis_object	analysis_object created from fine_tuning function.
summary	Whether to plot summary results table. Boolean (FALSE by default).
roc_curve	Whether to plot ROC Curve (Classification task only). Boolean (FALSE by default).
pr_curve	Whether to plot ROC Curve (Classification task only). Boolean (FALSE by default).
gain_curve	Whether to plot ROC Curve (Classification task only). Boolean (FALSE by default).
lift_curve	Whether to plot ROC Curve (Classification task only). Boolean (FALSE by default).
dist_by_class	Whether to plot distribution of output probability by class (Classification task only). Boolean (FALSE by default).
reliability_plot	Whether to plot Reliability Plot (Binary Classification task only). Boolean (FALSE by default).
confusion_matrix	Whether to Confusion Matrix (Classification task only). Boolean (FALSE by default).
scatter_residuals	Whether to plot Residuals vs Predictions (Regression task only). Boolean (FALSE by default).
scatter_predictions	Whether to plot Predictions vs Observed (Regression task only). Boolean (FALSE by default).
residuals_dist	Whether to plot Residuals Distribution (Regression task only). Boolean (FALSE by default).
new_data	Data to be used for Confusion Matrix, Reliability Plot, Distribution by Class Plot, Residuals vs Predictions Plot, Predictions vs Observed Plot and Residuals Distribution Plot. A string with the name of the data_set: "train", "validation", "test" (default) or "all".

**Value**

Updated analysis\_object

---

sim_data	<i>Example Data Set</i>
----------	-------------------------

---

**Description**

This dataset contains simulated data of a psychometric trial.

**Usage**

sim\_data

**Format**

A data frame with 1000 rows and 10 columns:

**psych\_well** Psychological Wellbeing Indicator. Continuous with (0,100)

**psych\_well\_bin** Psychological Wellbeing Binary Indicator. Factor with ("Low", "High")

**psych\_well\_pol** Psychological Wellbeing Polytomic Indicator. Factor with ("Low", "Somewhat", "Quite a bit", "Very Much")

**gender** Patient Gender. Factor ("Female", "Male")

**age** Patient Age. Continuous (18, 85)

**socioec\_status** Socioeconomial Status Indicator. Factor ("Low", "Medium", "High")

**emot\_intel** Emotional Intelligence Indicator. Continuous (24, 120)

**resilience** Resilience Indicator. Continuous (4, 20)

**depression** Depression Indicator. Continuous (0, 63)

**life\_sat** Life Satisfaction Indicator. Continuous (5, 35)



# Index

- \* **datasets**
  - sim\_data, [8](#)
- build\_model, [2](#)
- fine\_tuning, [4](#)
- preprocessing, [5](#)
- sensitivity\_analysis, [6](#)
- show\_results, [6](#)
- sim\_data, [8](#)