

Package ‘modelcardr’

July 30, 2026

Type Package

Title Machine Learning Prediction Auditing and Model Card Reporting

Version 0.1.0

Description Audits predictions from fitted machine learning models without requiring retraining or access to the fitted model object. It supports binary classification, multiclass classification, and regression, providing performance metrics with bootstrap confidence intervals, calibration and threshold analyses, residual diagnostics, subgroup comparisons, user-defined acceptance criteria, risk warnings, and self-contained reports. For binary-classification and regression models, the package can also assemble and render structured model cards documenting intended use, evaluation data, performance, subgroup results, assumptions, and limitations. The model-card reporting approach is described by Mitchell et al. (2019) ``Model Cards for Model Reporting" [<doi:10.1145/3287560.3287596>](https://doi.org/10.1145/3287560.3287596).

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports graphics, stats, tools, utils, withr

Suggests testthat (>= 3.0.0), jsonlite

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Georgios P. Georgiou [aut, cre]

Maintainer Georgios P. Georgiou <georgiou.georg@unic.ac.cy>

Repository CRAN

Date/Publication 2026-07-30 16:40:02 UTC

Contents

.audit_binary_predictions 2

add_data_description	4
add_limitations	5
add_purpose	5
audit_assumptions	6
audit_fairness	6
audit_performance	7
audit_predictions	7
audit_requirements	9
check_requirements	12
compare_cards	13
compare_predictions	13
model_card	14
model_card_from_predictions	16
plot.model_audit	17
print.audit_requirements	17
print.model_audit	18
print.model_card	18
print.prediction_comparison	19
render_audit	19
render_card	20
summary.model_audit	20
summary.model_card	21
validate_card	21

Index 22

.audit_binary_predictions

Audit predictions from a completed model

Description

Evaluates predictions without retraining or requiring the fitted model. For binary classification it calculates bootstrap confidence intervals, threshold trade-offs, ROC and precision-recall curves, probability calibration, subgroup performance, and automatic warning rules.

Usage

```
.audit_binary_predictions(
  truth,
  probability = NULL,
  estimate = NULL,
  positive = NULL,
  threshold = 0.5,
  thresholds = seq(0.05, 0.95, by = 0.01),
  groups = NULL,
  bootstrap = 500L,
  conf_level = 0.95,
```

```

    seed = NULL,
    threshold_objective = c("balanced_accuracy", "f1", "youden", "cost"),
    false_positive_cost = 1,
    false_negative_cost = 1,
    calibration_bins = 10L,
    reference_probability = NULL,
    drift_bins = 10L,
    subgroup_min_n = 30L,
    title = "Prediction Audit",
    requirements = NULL
  )

```

Arguments

<code>truth</code>	Observed outcomes.
<code>probability</code>	Numeric probability of the positive class.
<code>estimate</code>	Optional predicted classes. When omitted, classes are created from probability and threshold.
<code>positive</code>	Label of the positive class.
<code>threshold</code>	Threshold used for the main confusion matrix and metrics.
<code>thresholds</code>	Candidate thresholds to evaluate.
<code>groups</code>	Optional subgroup data frame, named list, or vector.
<code>bootstrap</code>	Number of stratified bootstrap samples for confidence intervals. Use zero to disable.
<code>conf_level</code>	Confidence level for bootstrap percentile intervals.
<code>seed</code>	Optional random seed used for bootstrap resampling. The default NULL uses the current random-number generator state without resetting it.
<code>threshold_objective</code>	Criterion for the recommended threshold: "balanced_accuracy", "f1", "youden", or "cost".
<code>false_positive_cost</code>	Relative cost of a false positive.
<code>false_negative_cost</code>	Relative cost of a false negative.
<code>calibration_bins</code>	Number of probability bins used in the calibration table.
<code>reference_probability</code>	Optional probabilities from an earlier or reference population for prediction-distribution drift checks.
<code>drift_bins</code>	Number of bins used for the population stability index.
<code>subgroup_min_n</code>	Subgroups below this size are flagged.
<code>title</code>	Report title.
<code>requirements</code>	Optional pass/fail criteria created by <code>audit_requirements()</code> .

Value

An object of class `model_audit`.

<code>add_data_description</code>	<i>Add data documentation</i>
-----------------------------------	-------------------------------

Description

Add data documentation

Usage

```
add_data_description(  
  card,  
  source = NULL,  
  collection_period = NULL,  
  population = NULL,  
  sampling = NULL,  
  exclusions = NULL,  
  preprocessing = NULL,  
  license = NULL,  
  notes = NULL  
)
```

Arguments

<code>card</code>	A model card.
<code>source</code>	Data source.
<code>collection_period</code>	Collection period.
<code>population</code>	Population represented by the data.
<code>sampling</code>	Sampling method.
<code>exclusions</code>	Exclusion criteria.
<code>preprocessing</code>	Important preprocessing steps.
<code>license</code>	Data license or access restrictions.
<code>notes</code>	Additional notes.

Value

The updated card.

add_limitations	<i>Add known limitations</i>
-----------------	------------------------------

Description

Add known limitations

Usage

```
add_limitations(card, ...)
```

Arguments

- | | |
|------|---|
| card | A model card. |
| ... | Character vectors containing limitations. |

Value

The updated card.

add_purpose	<i>Add intended-use information</i>
-------------	-------------------------------------

Description

Add intended-use information

Usage

```
add_purpose(  
  card,  
  intended_use,  
  users = NULL,  
  out_of_scope = NULL,  
  decision_context = NULL,  
  owner = NULL  
)
```

Arguments

- | | |
|------------------|---|
| card | A model card. |
| intended_use | Intended use of the model. |
| users | Intended users. |
| out_of_scope | Unsupported or prohibited uses. |
| decision_context | How predictions enter a decision process. |
| owner | Responsible person or team. |

Value

The updated card.

audit_assumptions	<i>Run model-specific diagnostic checks</i>
-------------------	---

Description

Diagnostics are screening evidence, not certification that assumptions hold.

Usage

```
audit_assumptions(card)
```

Arguments

card	A model card.
------	---------------

Value

The updated card.

audit_fairness	<i>Audit subgroup performance and fairness</i>
----------------	--

Description

This function reports subgroup metrics and disparities. It does not declare a model to be fair or unfair; interpretation depends on the use case.

Usage

```
audit_fairness(card, groups = NULL, min_n = 30L)
```

Arguments

card	A model card.
groups	Optional character vector naming stored sensitive features. Defaults to all sensitive features supplied to <code>model_card()</code> .
min_n	Minimum subgroup size before a subgroup is marked as small.

Value

The updated card.

audit_performance	<i>Calculate model performance</i>
-------------------	------------------------------------

Description

Calculate model performance

Usage

```
audit_performance(card)
```

Arguments

card	A model card.
------	---------------

Value

The updated card.

audit_predictions	<i>Audit Predictions from a Completed Model</i>
-------------------	---

Description

Audits completed binary-classification, multiclass-classification, or regression predictions without requiring the fitted model.

Usage

```
audit_predictions(  
  truth,  
  probability = NULL,  
  estimate = NULL,  
  task = c("auto", "binary_classification", "multiclass_classification", "regression"),  
  positive = NULL,  
  class_names = NULL,  
  threshold = 0.5,  
  thresholds = seq(0.05, 0.95, by = 0.01),  
  groups = NULL,  
  bootstrap = 500L,  
  conf_level = 0.95,  
  seed = NULL,  
  threshold_objective = c("balanced_accuracy", "f1", "youden", "cost"),  
  false_positive_cost = 1,  
  false_negative_cost = 1,  
  calibration_bins = 10L,
```

```

    reference_probability = NULL,
    drift_bins = 10L,
    subgroup_min_n = 30L,
    title = "Prediction Audit",
    requirements = NULL
)

```

Arguments

<code>truth</code>	Observed outcomes.
<code>probability</code>	A binary positive-class probability vector or multiclass probability matrix.
<code>estimate</code>	Predicted classes or numeric regression estimates.
<code>task</code>	Task type, or "auto".
<code>positive</code>	Positive class for binary classification.
<code>class_names</code>	Optional multiclass probability-column names.
<code>threshold</code>	Binary decision threshold.
<code>thresholds</code>	Candidate binary thresholds.
<code>groups</code>	Optional subgroup variables.
<code>bootstrap</code>	Number of bootstrap samples.
<code>conf_level</code>	Bootstrap confidence level.
<code>seed</code>	Optional random seed used for bootstrap resampling. The default NULL uses the current random-number generator state without resetting it.
<code>threshold_objective</code>	Binary threshold objective.
<code>false_positive_cost</code>	Relative binary false-positive cost.
<code>false_negative_cost</code>	Relative binary false-negative cost.
<code>calibration_bins</code>	Number of calibration bins.
<code>reference_probability</code>	Optional binary reference probabilities for drift.
<code>drift_bins</code>	Number of drift bins.
<code>subgroup_min_n</code>	Minimum subgroup size before a warning.
<code>title</code>	Audit title.
<code>requirements</code>	Optional criteria from <code>audit_requirements()</code> .

Value

An object of class `model_audit`.

Examples

```
audit_predictions(c(1, 2, 3), estimate = c(1.1, 1.9, 3.2), task = "regression", bootstrap = 0)
```

audit_requirements	<i>Define prediction-audit requirements</i>
--------------------	---

Description

Creates optional acceptance criteria for binary classification, multiclass classification, and regression audits. Requirements that cannot be evaluated from the supplied data are reported as INCONCLUSIVE.

Usage

```
audit_requirements(
  min_n = NULL,
  min_accuracy = NULL,
  min_balanced_accuracy = NULL,
  min_sensitivity = NULL,
  min_specificity = NULL,
  min_precision = NULL,
  min_f1 = NULL,
  min_roc_auc = NULL,
  min_average_precision = NULL,
  min_macro_f1 = NULL,
  min_weighted_f1 = NULL,
  min_macro_roc_auc = NULL,
  min_minimum_class_recall = NULL,
  min_r_squared = NULL,
  max_false_positive_rate = NULL,
  max_false_negative_rate = NULL,
  max_brier_score = NULL,
  max_log_loss = NULL,
  max_multiclass_log_loss = NULL,
  max_multiclass_brier_score = NULL,
  max_rmse = NULL,
  max_mae = NULL,
  max_absolute_bias = NULL,
  max_mape = NULL,
  max_smape = NULL,
  max_expected_calibration_error = NULL,
  calibration_slope_range = NULL,
  max_threshold_difference = NULL,
  max_subgroup_sensitivity_gap = NULL,
  max_subgroup_specificity_gap = NULL,
  max_subgroup_selection_rate_gap = NULL,
  max_subgroup_accuracy_gap = NULL,
  max_subgroup_macro_f1_gap = NULL,
  max_subgroup_rmse_gap = NULL,
  max_subgroup_mae_gap = NULL,
  max_population_stability_index = NULL,
```

```

    use_confidence_bounds = TRUE
)

```

Arguments

<code>min_n</code>	Minimum required number of evaluation observations. Use NULL to disable this requirement.
<code>min_accuracy</code>	Minimum acceptable classification accuracy. Use NULL to disable this requirement.
<code>min_balanced_accuracy</code>	Minimum acceptable balanced accuracy. Use NULL to disable this requirement.
<code>min_sensitivity</code>	Minimum acceptable binary-classification sensitivity. Use NULL to disable this requirement.
<code>min_specificity</code>	Minimum acceptable binary-classification specificity. Use NULL to disable this requirement.
<code>min_precision</code>	Minimum acceptable binary-classification precision. Use NULL to disable this requirement.
<code>min_f1</code>	Minimum acceptable binary-classification F1 score. Use NULL to disable this requirement.
<code>min_roc_auc</code>	Minimum acceptable binary-classification ROC AUC. Use NULL to disable this requirement.
<code>min_average_precision</code>	Minimum acceptable average precision for binary classification. Use NULL to disable this requirement.
<code>min_macro_f1</code>	Minimum acceptable macro F1 score for multiclass classification. Use NULL to disable this requirement.
<code>min_weighted_f1</code>	Minimum acceptable weighted F1 score for multiclass classification. Use NULL to disable this requirement.
<code>min_macro_roc_auc</code>	Minimum acceptable macro ROC AUC for multiclass classification. Use NULL to disable this requirement.
<code>min_minimum_class_recall</code>	Minimum acceptable recall for the worst-performing class in multiclass classification. Use NULL to disable this requirement.
<code>min_r_squared</code>	Minimum acceptable R-squared value for regression. Use NULL to disable this requirement.
<code>max_false_positive_rate</code>	Maximum acceptable false-positive rate for binary classification. Use NULL to disable this requirement.
<code>max_false_negative_rate</code>	Maximum acceptable false-negative rate for binary classification. Use NULL to disable this requirement.

max_brier_score	Maximum acceptable Brier score for binary classification. Use NULL to disable this requirement.
max_log_loss	Maximum acceptable log loss for binary classification. Use NULL to disable this requirement.
max_multiclass_log_loss	Maximum acceptable log loss for multiclass classification. Use NULL to disable this requirement.
max_multiclass_brier_score	Maximum acceptable Brier score for multiclass classification. Use NULL to disable this requirement.
max_rmse	Maximum acceptable root mean squared error for regression. Use NULL to disable this requirement.
max_mae	Maximum acceptable mean absolute error for regression. Use NULL to disable this requirement.
max_absolute_bias	Maximum acceptable absolute mean bias for regression. Use NULL to disable this requirement.
max_mape	Maximum acceptable mean absolute percentage error for regression. Use NULL to disable this requirement.
max_smape	Maximum acceptable symmetric mean absolute percentage error for regression. Use NULL to disable this requirement.
max_expected_calibration_error	Maximum acceptable expected calibration error for classification. Use NULL to disable this requirement.
calibration_slope_range	Numeric vector of length two containing the minimum and maximum acceptable binary calibration slope. Use NULL to disable this requirement.
max_threshold_difference	Maximum acceptable absolute difference between the selected binary threshold and the package-recommended threshold. Use NULL to disable this requirement.
max_subgroup_sensitivity_gap	Maximum acceptable difference between the highest and lowest subgroup sensitivities. Use NULL to disable this requirement.
max_subgroup_specificity_gap	Maximum acceptable difference between the highest and lowest subgroup specificities. Use NULL to disable this requirement.
max_subgroup_selection_rate_gap	Maximum acceptable difference between the highest and lowest subgroup selection rates. Use NULL to disable this requirement.
max_subgroup_accuracy_gap	Maximum acceptable difference between the highest and lowest subgroup accuracies. Use NULL to disable this requirement.
max_subgroup_macro_f1_gap	Maximum acceptable difference between the highest and lowest subgroup macro F1 scores. Use NULL to disable this requirement.

`max_subgroup_rmse_gap`
Maximum acceptable difference between the highest and lowest subgroup RMSE values. Use NULL to disable this requirement.

`max_subgroup_mae_gap`
Maximum acceptable difference between the highest and lowest subgroup MAE values. Use NULL to disable this requirement.

`max_population_stability_index`
Maximum acceptable population stability index for prediction drift. Use NULL to disable this requirement.

`use_confidence_bounds`
Logical. When TRUE, minimum requirements use lower confidence bounds and maximum requirements use upper confidence bounds when bootstrap intervals are available. When FALSE, point estimates are used.

Value

An object of class `audit_requirements` containing the configured audit criteria.

Examples

```
rules <- audit_requirements(
  min_n = 100,
  min_accuracy = 0.75,
  min_roc_auc = 0.80,
  max_log_loss = 0.60
)

print(rules)
```

<code>check_requirements</code>	<i>Apply pass/fail requirements to a prediction audit</i>
---------------------------------	---

Description

Evaluates configured acceptance criteria and adds a rule table plus an overall PASS, FAIL, or INCONCLUSIVE result to a model audit. Missing evidence is marked INCONCLUSIVE rather than passing automatically.

Usage

```
check_requirements(audit, requirements)
```

Arguments

`audit` An object of class `model_audit`.

`requirements` An object created by `audit_requirements()`.

Value

The updated model_audit object.

Examples

```
truth <- c("no", "no", "yes", "yes")
probability <- c(0.1, 0.2, 0.8, 0.9)
audit <- audit_predictions(truth, probability, positive = "yes", bootstrap = 0)
rules <- audit_requirements(min_roc_auc = 0.80, min_sensitivity = 0.80)
audit <- check_requirements(audit, rules)
audit$overall_status
audit$rule_results
```

compare_cards	<i>Compare two model cards</i>
---------------	--------------------------------

Description

Compare two model cards

Usage

```
compare_cards(old, new)
```

Arguments

- old Older model card.
- new Newer model card.

Value

A list containing metadata and metric comparisons.

compare_predictions	<i>Compare two sets of binary-classification predictions</i>
---------------------	--

Description

Uses paired stratified bootstrap resampling so each model is evaluated on the same resampled observations.

Usage

```
compare_predictions(  
  truth,  
  probability_a,  
  probability_b,  
  positive = NULL,  
  threshold = 0.5,  
  bootstrap = 1000L,  
  conf_level = 0.95,  
  seed = NULL,  
  model_names = c("model_a", "model_b")  
)
```

Arguments

truth	Observed outcomes.
probability_a	Positive-class probabilities from model A.
probability_b	Positive-class probabilities from model B.
positive	Positive-class label.
threshold	Threshold applied to both models.
bootstrap	Number of paired bootstrap samples.
conf_level	Confidence level for differences.
seed	Optional random seed used for paired bootstrap resampling. The default NULL uses the current random-number generator state without resetting it.
model_names	Two model names.

Value

An object of class prediction_comparison.

model_card	Create a model card
------------	---------------------

Description

Creates a structured model-card object for a regression or binary classification model. Predictions may be generated from model, or supplied directly through estimate and probability.

Usage

```
model_card(
  model = NULL,
  evaluation_data = NULL,
  truth,
  estimate = NULL,
  probability = NULL,
  task = c("auto", "regression", "binary_classification"),
  positive = NULL,
  threshold = 0.5,
  title = NULL,
  version = "0.1.0",
  sensitive_features = NULL,
  metadata = list()
)
```

Arguments

<code>model</code>	A fitted model object, or NULL for prediction-only cards.
<code>evaluation_data</code>	A data frame used for evaluation.
<code>truth</code>	Outcome vector or the name of its column in <code>evaluation_data</code> .
<code>estimate</code>	Predicted values/classes or a column name.
<code>probability</code>	Positive-class probabilities or a column name.
<code>task</code>	One of "auto", "regression", "binary_classification", or "multiclass_classification".
<code>positive</code>	Positive outcome label for classification.
<code>threshold</code>	Probability threshold used to create class predictions.
<code>title</code>	Card title.
<code>version</code>	Model/card version.
<code>sensitive_features</code>	Character vector of columns to retain for subgroup evaluation. Raw predictor columns are not stored.
<code>metadata</code>	Named list of extra metadata.

Value

An object of class `model_card`.

model_card_from_predictions

Compatibility Interface for Prediction Audits

Description

Calls `audit_predictions()` for all supported prediction tasks.

Usage

```
model_card_from_predictions(
  truth,
  estimate = NULL,
  probability = NULL,
  task = c("auto", "regression", "binary_classification", "multiclass_classification"),
  positive = NULL,
  class_names = NULL,
  threshold = 0.5,
  title = "Prediction Audit",
  groups = NULL,
  bootstrap = 500L,
  ...
)
```

Arguments

<code>truth</code>	Observed outcomes.
<code>estimate</code>	Predicted classes or numeric estimates.
<code>probability</code>	Binary probability vector or multiclass probability matrix.
<code>task</code>	Task type.
<code>positive</code>	Binary positive class.
<code>class_names</code>	Optional multiclass probability-column names.
<code>threshold</code>	Binary classification threshold.
<code>title</code>	Audit title.
<code>groups</code>	Optional subgroup variables.
<code>bootstrap</code>	Bootstrap samples.
<code>...</code>	Additional arguments passed to <code>audit_predictions()</code> .

Value

A `model_audit` object.

plot.model_audit	<i>Plot a prediction audit</i>
------------------	--------------------------------

Description

Plot a prediction audit

Usage

```
## S3 method for class 'model_audit'
plot(x, type = NULL, metric = "balanced_accuracy", ...)
```

Arguments

x	A model_audit object.
type	Plot type. Binary audits support "roc", "pr", "calibration", and "threshold"; multiclass audits support "calibration"; regression audits support "actual_predicted", "residuals", and "residual_histogram".
metric	Metric shown for a binary threshold plot.
...	Additional graphical arguments.

Value

The input model_audit object, returned invisibly. The function is called for its side effect of drawing the selected diagnostic plot.

print.audit_requirements	<i>Print audit requirements</i>
--------------------------	---------------------------------

Description

Print audit requirements

Usage

```
## S3 method for class 'audit_requirements'
print(x, ...)
```

Arguments

x	An audit_requirements object.
...	Unused.

Value

The input `audit_requirements` object, returned invisibly. The function is called for its side effect of printing the configured audit criteria and whether confidence bounds are used.

<code>print.model_audit</code>	<i>Print a prediction audit</i>
--------------------------------	---------------------------------

Description

Print a prediction audit

Usage

```
## S3 method for class 'model_audit'
print(x, ...)
```

Arguments

<code>x</code>	A <code>model_audit</code> object.
<code>...</code>	Unused.

Value

The input `model_audit` object, returned invisibly. The function is called for its side effect of printing a concise overview of the audit, including its task, sample size, selected metrics, status, and number of findings.

<code>print.model_card</code>	<i>Print a model card</i>
-------------------------------	---------------------------

Description

Print a model card

Usage

```
## S3 method for class 'model_card'
print(x, ...)
```

Arguments

<code>x</code>	A <code>model_card</code> object.
<code>...</code>	Unused.

Value

The input model_card object, returned invisibly. The function is called for its side effect of printing the card title, version, task, model class, evaluation sample size, and number of performance metrics.

```
print.prediction_comparison
```

Print a prediction comparison

Description

Print a prediction comparison

Usage

```
## S3 method for class 'prediction_comparison'
print(x, ...)
```

Arguments

x	A prediction_comparison object.
...	Unused.

Value

The input prediction_comparison object, returned invisibly. The function is called for its side effect of printing the compared model names, evaluation sample size, and selected comparison metrics.

render_audit	<i>Render a Prediction Audit</i>
--------------	----------------------------------

Description

Renders task-specific evidence and omits sections that do not apply.

Usage

```
render_audit(audit, output_file, format = NULL, detailed = FALSE)
```

Arguments

audit	A model_audit object.
output_file	Destination HTML, Markdown, or JSON file.
format	Optional explicit output format.
detailed	Whether to include detailed diagnostic tables.

Value

The output path, invisibly.

render_card	<i>Render a model card</i>
-------------	----------------------------

Description

Render a model card

Usage

```
render_card(card, output_file, format = NULL)
```

Arguments

- card A model card.
- output_file Destination file ending in .md, .html, or .json.
- format Optional explicit format: "markdown", "html", or "json".

Value

The normalized output path, invisibly.

summary.model_audit	<i>Summarize a prediction audit</i>
---------------------	-------------------------------------

Description

Summarize a prediction audit

Usage

```
## S3 method for class 'model_audit'  
summary(object, ...)
```

Arguments

- object A model_audit object.
- ... Unused.

Value

A named list summarizing the audit. It contains the audit title, task, evaluation-data information, overall and class-level metrics, recommended threshold, calibration summary, subgroup disparities, diagnostics, requirement-checking results, findings, and risk warnings. Components that are not applicable to the audit task may be NULL.

summary.model_card	<i>Summarize a model card</i>
--------------------	-------------------------------

Description

Summarize a model card

Usage

```
## S3 method for class 'model_card'  
summary(object, ...)
```

Arguments

object	A model_card object.
...	Unused.

Value

A named list containing the model-card title, version, task, model metadata, intended purpose, performance results, fairness summary, assumption checks, limitations, and completeness-validation results.

validate_card	<i>Validate model-card completeness</i>
---------------	---

Description

Validate model-card completeness

Usage

```
validate_card(card)
```

Arguments

card	A model card.
------	---------------

Value

A data frame of validation checks.

Index

`.audit_binary_predictions`, [2](#)

`add_data_description`, [4](#)
`add_limitations`, [5](#)
`add_purpose`, [5](#)
`audit_assumptions`, [6](#)
`audit_fairness`, [6](#)
`audit_performance`, [7](#)
`audit_predictions`, [7](#)
`audit_requirements`, [9](#)

`check_requirements`, [12](#)
`compare_cards`, [13](#)
`compare_predictions`, [13](#)

`model_card`, [14](#)
`model_card()`, [6](#)
`model_card_from_predictions`, [16](#)

`plot.model_audit`, [17](#)
`print.audit_requirements`, [17](#)
`print.model_audit`, [18](#)
`print.model_card`, [18](#)
`print.prediction_comparison`, [19](#)

`render_audit`, [19](#)
`render_card`, [20](#)

`summary.model_audit`, [20](#)
`summary.model_card`, [21](#)

`validate_card`, [21](#)