

# Package ‘sensiverse’

November 6, 2025

**Type** Package

**Title** Sensitivity and Model Uncertainty Analysis

**Version** 0.0.1

**Description** Tools to estimate large model spaces, analyze uncertainty via multinomial logit, neural nets, and neighbour changes. The package is based on the PNAS article Ganslmeier and Vlandas (2025): Estimating the extent and sources of model uncertainty in political science

**License** MIT + file LICENSE

**Imports** dplyr, purrr, tibble, stringr, ggplot2, lmtest, sandwich, multiwayvcov, nnet, caret, pbapply

**Suggests** keras, tensorflow, vip, iml, testthat, knitr, rmarkdown

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 7.3.1

**VignetteBuilder** knitr

**URL** <https://www.pnas.org/doi/10.1073/pnas.2414926122>

**BugReports** <https://github.com/MGanslmeier/sensiverse/issues>

## R topics documented:

calculate_sign_shares . . . . .	2
estimate_model_space . . . . .	2
filter_model_space . . . . .	3
find_uncertainty_source_mlogit . . . . .	3
find_uncertainty_source_neigh . . . . .	4
find_uncertainty_source_neuronet . . . . .	5
plot_importance_mlogit . . . . .	6
plot_importance_neigh . . . . .	6
plot_importance_neuronet . . . . .	7
plot_sign_share . . . . .	7

**Index**

8

---

`calculate_sign_shares` *Compute sign shares for the focus variable*

---

## Description

Aggregates the distribution of `sig_sign` across all specs or by a given dimension (e.g., `"fes"`, `"set"`, `"sample"`, `"dep"`).

## Usage

```
calculate_sign_shares(est_tbl, dimension = NULL)
```

## Arguments

<code>est_tbl</code>	Tibble from <code>estimate_model_space()</code> (or a filtered subset).
<code>dimension</code>	Optional character. If provided, compute shares by this column; if <code>NULL</code> , returns overall shares.

## Value

Tibble with counts, total, and signed percentages (`perc`). Negative values reflect negative significant shares.

## See Also

Other model-space: [estimate\\_model\\_space\(\)](#), [filter\\_model\\_space\(\)](#), [plot\\_sign\\_share\(\)](#)

---

`estimate_model_space` *Estimate a specification universe (OLS) for a focus variable*

---

## Description

Runs OLS regressions across a user-defined grid (dependent variables, control sets, fixed effects, SE type, and sample split) and collects results only for `focus_var`.

## Usage

```
estimate_model_space(df, focus_var, specs, space_n = 150000)
```

## Arguments

<code>df</code>	Data frame containing <code>DEP*</code> , <code>CONT*</code> , <code>FEgroup</code> , <code>FETIME</code> , and <code>SAMPLE*</code> columns.
<code>focus_var</code>	Character. IV of interest (e.g., <code>"CONTa"</code> ).
<code>specs</code>	Named list with components:
<code>dep</code>	Character vector of dependent variables present in <code>df</code> .
<code>cont</code>	Character vector of candidate controls (e.g., <code>CONT*</code> ).
<code>fes</code>	Subset of <code>c("none", "time", "group", "time+group")</code> .
<code>set</code>	Subset of <code>c("simple", "robust", "clustered")</code> .
<code>sample</code>	Character vector of sample-split dummies (cols in <code>df</code> ).
<code>space_n</code>	Optional integer. If the full grid is larger, a random sample of at most <code>space_n</code> specifications is estimated (for speed).

**Value**

A tibble with coefficient, SE, p-value, significance class, and full specification metadata for `focus_var`.

**See Also**

Other model-space: `calculate_sign_shares()`, `filter_model_space()`, `plot_sign_share()`

---

`filter_model_space`      *Filter a model-space result*

---

**Description**

Post-process the output of `estimate_model_space()` to retain a subset based on AIC quantiles, forbidden pairs, or an arbitrary logical expression evaluated on the result.

**Usage**

```
filter_model_space(res, top_aic_pct = NULL, drop_expr = NULL)
```

**Arguments**

<code>res</code>	Tibble from <code>estimate_model_space()</code> .
<code>top_aic_pct</code>	Numeric in [0,1]. Keep the best (lowest) <code>top_aic_pct</code> share by AIC. If <code>NULL</code> , do not filter by AIC.
<code>drop_expr</code>	Character string, an expression evaluated in the context of <code>res</code> (e.g., <code>"dep == 'DEP1' &amp; grepl('CONTb', contset)"</code> ).

**Value**

Filtered tibble (same schema as `res`).

**See Also**

Other model-space: `calculate_sign_shares()`, `estimate_model_space()`, `plot_sign_share()`

---

`find_uncertainty_source_mlogit`      *Identify uncertainty sources via mlogit*

---

**Description**

Trains `nnet::multinom` on the MLOGIT source dataset and evaluates how changing each spec feature shifts predicted probabilities across the three outcome classes.

**Usage**

```
find_uncertainty_source_mlogit(res, focus)
```

**Arguments**

<b>res</b>	Tibble from <code>estimate_model_space()</code> .
<b>focus</b>	Character. Focus variable (matches IV).

**Value**

A list with elements:

**method** "MLOGIT"  
**focus** Focus variable  
**model** Fitted `nnet::multinom`  
**pp\_change** Data frame of probability changes

**See Also**

Other estimate-sources: [find\\_uncertainty\\_source\\_neigh\(\)](#), [find\\_uncertainty\\_source\\_neuronet\(\)](#)

**find\_uncertainty\_source\_neigh**

*Identify uncertainty sources via nearest neighbour approach*

**Description**

Monte-Carlo heuristic: sample a baseline specification, flip exactly one dimension (dep/fes/set/sample/cont) to its nearest neighbour, re-estimate, and record whether the significance class changes for focus.

**Usage**

```
find_uncertainty_source_neigh(
  specs,
  df,
  focus,
  n_draws = 100,
  seed = 123,
  max_iter = 10 * n_draws
)
```

**Arguments**

<b>specs</b>	The same specs list used in <code>estimate_model_space()</code> .
<b>df</b>	Original data used for estimation.
<b>focus</b>	Character. Focus variable.
<b>n_draws</b>	Integer. Number of perturbation draws.
<b>seed</b>	Random seed.
<b>max_iter</b>	Safety cap on attempts to complete <code>n_draws</code> .

**Value**

Data frame with one row per draw and flip indicator.

## See Also

Other estimate-sources: [find\\_uncertainty\\_source\\_mlogit\(\)](#), [find\\_uncertainty\\_source\\_neuronet\(\)](#)

---

`find_uncertainty_source_neuronet`

*Identify uncertainty sources via neural network approach*

---

## Description

Trains separate one-vs-rest neural networks for each outcome class ("not significant", "significant negative", "significant positive") and computes SHAP importances (via `vip`), and aggregates to both features and dimensions.

## Usage

```
find_uncertainty_source_neuronet(  
  res,  
  focus,  
  grid = list(dropout = c(0, 0.3), layers = c(3, 5), units = c(32, 64), learning_rate =  
    c(0.001)),  
  split = c(train = 0.6, val = 0.2, test = 0.2),  
  epochs = 15,  
  batch_size = 256,  
  seed = 123,  
  verbose = 0  
)
```

## Arguments

<code>res</code>	Tibble from <code>estimate_model_space()</code> .
<code>focus</code>	Character. Focus variable.
<code>grid</code>	List of hyperparameters with elements <code>dropout</code> , <code>layers</code> , <code>units</code> , <code>learning_rate</code> . A small grid is recommended.
<code>split</code>	Named numeric vector with <code>train</code> , <code>val</code> , <code>test</code> shares.
<code>epochs</code>	Integer training epochs.
<code>batch_size</code>	Mini-batch size.
<code>seed</code>	Random seed.
<code>verbose</code>	Keras verbosity (0/1/2).

## Value

List with method label, focus, and per-class SHAP summaries:

`all_signs` Named list (per outcome) with `shap_feature` and `shap_dimension`.

## See Also

Other estimate-sources: [find\\_uncertainty\\_source\\_mlogit\(\)](#), [find\\_uncertainty\\_source\\_neigh\(\)](#)

**plot\_importance\_mlogit**

*Plot feature importance for mlogit approach*

## Description

Visual summary of average absolute probability changes by dimension or by specific feature/value when using the MLOGIT method.

## Usage

```
plot_importance_mlogit(out, aggregate = TRUE)
```

## Arguments

- |           |  |
|-----------|--|
| out       | Output list from <code>find_uncertainty_source_mlogit()</code> . |
| aggregate | Logical. If TRUE, aggregates to dimensions.                      |

## Value

A **ggplot2** object.

## See Also

Other plot-sources: [plot\\_importance\\_neigh\(\)](#), [plot\\_importance\\_neuronet\(\)](#)

**plot\_importance\_neigh** *Plot flip rates by dimension using nearest-neighbour approach*

## Description

Shows how often a one-step spec perturbation flips the significance class of the focus variable—summarized by dimension.

## Usage

```
plot_importance_neigh(out)
```

## Arguments

- |     |  |
|-----|--|
| out | Data frame from <code>find_uncertainty_source_neigh()</code> . |
|-----|--|

## Value

A **ggplot2** object.

## See Also

Other plot-sources: [plot\\_importance\\_mlogit\(\)](#), [plot\\_importance\\_neuronet\(\)](#)

---

```
plot_importance_neuronet
```

*Plot importances (SHAP) of neural network approach*

---

## Description

Displays either dimension-level average SHAP shares (aggregate = TRUE) or feature-level SHAP shares (aggregate = FALSE), split by outcome class.

## Usage

```
plot_importance_neuronet(out, aggregate = TRUE)
```

## Arguments

out	Output of <code>find_uncertainty_source_neuronet()</code> .
aggregate	Logical. If TRUE, plot dimension aggregates.

## Value

A `ggplot2` object.

## See Also

Other plot-sources: [plot\\_importance\\_mlogit\(\)](#), [plot\\_importance\\_neigh\(\)](#)

---

---

```
plot_sign_share
```

*Plot sign shares of the model space*

---

## Description

Visualizes signed shares from `calculate_sign_shares()` either overall or by dimension

## Usage

```
plot_sign_share(shares, dimension = NULL)
```

## Arguments

shares	Output of <code>calculate_sign_shares()</code> .
dimension	Character or NULL. If grouped by dimension, pass the same value used in <code>calculate_sign_shares()</code> for consistent axis labels.

## Value

A `ggplot2` object.

## See Also

Other model-space: [calculate\\_sign\\_shares\(\)](#), [estimate\\_model\\_space\(\)](#), [filter\\_model\\_space\(\)](#)

# Index

- \* **estimate-sources**
  - find\_uncertainty\_source\_mlogit, 3
  - find\_uncertainty\_source\_neigh, 4
  - find\_uncertainty\_source\_neuronet,  
5
- \* **model-space**
  - calculate\_sign\_shares, 2
  - estimate\_model\_space, 2
  - filter\_model\_space, 3
  - plot\_sign\_share, 7
- \* **plot-sources**
  - plot\_importance\_mlogit, 6
  - plot\_importance\_neigh, 6
  - plot\_importance\_neuronet, 7
- calculate\_sign\_shares, 2, 3, 7
- estimate\_model\_space, 2, 2, 3, 7
- filter\_model\_space, 2, 3, 3, 7
- find\_uncertainty\_source\_mlogit, 3, 5
- find\_uncertainty\_source\_neigh, 4, 4, 5
- find\_uncertainty\_source\_neuronet, 4, 5,  
5
- plot\_importance\_mlogit, 6, 6, 7
- plot\_importance\_neigh, 6, 6, 7
- plot\_importance\_neuronet, 6, 7
- plot\_sign\_share, 2, 3, 7