

Package: trama.ml (via r-universe)

October 10, 2026

Title Interpretable and Classical Machine Learning for trama

Version 0.5.8

Description Supervised learning nodes for trama workflows, from linear baselines and interpretable CART and FIGS models to random forests, support vector machines and gradient boosted trees. Provides reproducible training and test splits, prediction, evaluation and model inspection. Learning engines are optional and loaded only for the selected model. Neural networks are deliberately outside the scope of this collection.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.1)

Imports trama (>= 0.5.10), trama.data (>= 0.4.0), trama.view, trama.models, tibble, rlang, cli (>= 3.6.0), ggplot2, stats, utils, datasets

Suggests figsr (>= 0.1.1), rpart, ranger, yardstick, PRROC, pROC, e1071, xgboost (>= 3.0.0), testthat (>= 3.0.0), pkgload

Config/testthat/edition 3

Config/trama/requires trama.data, trama.view, trama.models

Remotes uaipetro/trama,
trama.data=github::uaipetro/trama/collections/trama.data,
trama.view=github::uaipetro/trama/collections/trama.view,
trama.models=github::uaipetro/trama/collections/trama.models

Config/roxygen2/version 8.0.0

Config/pak/sysreqs
cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://uaipetro.r-universe.dev>

Date/Publication 2026-10-09 16:23:25 UTC

RemoteUrl <https://github.com/uaipetro/trama>

RemoteRef HEAD
RemoteSha 1bdeb4f693df836f77e762f2642cc18a3956f8b1
RemoteSubdir collections/trama.ml

Contents

tr_ml_cart	2
tr_ml_example	3
tr_ml_figs	4
tr_ml_fit	4
tr_ml_forest	6
tr_ml_linear	7
tr_ml_nested_cv	8
tr_ml_residuals	10
tr_ml_rules	11
tr_ml_split	11
tr_ml_svm	12
tr_ml_tree_plot	13
tr_ml_tune	14
tr_ml_tuning_plot	15
tr_ml_xgboost	16
trama_collection	17
Index	18

tr_ml_cart	<i>Árvore CART para regressão ou classificação.</i>
------------	---

Description

Árvore CART para regressão ou classificação.

Usage

```
tr_ml_cart(  
  dados,  
  resposta = "",  
  preditores = "",  
  tarefa = "auto",  
  max_depth = 3L,  
  min_n = 5L,  
  cp = 0,  
  poda = "1ep",  
  seed = 42L  
)
```

Arguments

dados	Tabela de treino.
resposta	Nome da coluna a prever.
preditores	Preditores separados por vírgula; em branco, usa todas as colunas numéricas, exceto o resposta.
tarefa	"auto", "regressao" ou "classificacao".
max_depth	Profundidade máxima das árvores (CART, forest e XGBoost).
min_n	Mínimo por folha no CART; tamanho mínimo do nó em FIGS e ranger.
cp	Parâmetro de complexidade do CART: a árvore cresce só com divisões que melhoram o ajuste relativo em pelo menos cp; zero cresce a árvore máxima permitida por max_depth e min_n, depois podada.
poda	Poda do CART por custo-complexidade, escolhida pela validação cruzada do rpart com min(10, n) folds (Breiman et al. 1984): "1ep" fica com a menor árvore cujo erro de validação não passa do mínimo mais um erro-padrão; "minimo", com a de menor erro; "nenhuma" não poda.
seed	Semente local; o estado aleatório da sessão é preservado.

Value

Um modelo tr_ml_fit.

tr_ml_example

Carrega um conjunto pequeno para exemplos da coleção ML

Description

Carrega um conjunto pequeno para exemplos da coleção ML

Usage

```
tr_ml_example(nome = "iris")
```

Arguments

nome	iris, iris_binaria ou mtcars.
------	-------------------------------

Value

Um tibble.

tr_ml_figs

*Soma de árvores interpretáveis FIGS.***Description**

Soma de árvores interpretáveis FIGS.

Usage

```
tr_ml_figs(
  dados,
  resposta = "",
  preditores = "",
  tarefa = "auto",
  max_splits = 6L,
  min_n = 5L,
  seed = 42L
)
```

Arguments

dados	Tabela de treino.
resposta	Nome da coluna a prever.
preditores	Preditores separados por vírgula; em branco, usa todas as colunas numéricas, exceto o resposta.
tarefa	"auto", "regressao" ou "classificacao".
max_splits	Orçamento total de divisões do FIGS.
min_n	Mínimo por folha no CART; tamanho mínimo do nó em FIGS e ranger.
seed	Semente local; o estado aleatório da sessão é preservado.

Value

Um modelo tr_ml_fit.

tr_ml_fit

*Ajustar um modelo clássico de aprendizado de máquina***Description**

Ajusta regressão linear/logística, CART, FIGS, floresta aleatória, SVM ou XGBoost sobre preditores numéricos. Os engines além de stats são opcionais e a mensagem de erro informa exatamente qual pacote instalar.

Usage

```
tr_ml_fit(  
  dados,  
  resposta = "",  
  preditores = "",  
  modelo = "cart",  
  tarefa = "auto",  
  seed = 42L,  
  max_depth = 3L,  
  min_n = 5L,  
  max_splits = 6L,  
  trees = 200L,  
  mtry = 0L,  
  cost = 1,  
  gamma = 0.1,  
  kernel = "radial",  
  nrounds = 100L,  
  eta = 0.1,  
  cp = 0,  
  poda = "1ep",  
  corte = 0.5,  
  importancia = "impureza"  
)
```

Arguments

dados	Tabela de treino.
resposta	Nome da coluna a prever.
preditores	Preditores separados por vírgula; em branco, usa todas as colunas numéricas, exceto o resposta.
modelo	Um de "linear", "cart", "figs", "forest", "svm" ou "xgboost".
tarefa	"auto", "regressao" ou "classificacao".
seed	Semente local; o estado aleatório da sessão é preservado.
max_depth	Profundidade máxima das árvores (CART, forest e XGBoost).
min_n	Mínimo por folha no CART; tamanho mínimo do nó em FIGS e ranger.
max_splits	Orçamento total de divisões do FIGS.
trees	Número de árvores da floresta.
mtry	Preditores candidatos por divisão; zero usa raiz de p.
cost	Penalidade dos erros da SVM, estritamente positiva.
gamma	Escala do kernel da SVM.
kernel	Kernel linear, radial, polynomial ou sigmoid.
nrounds	Rodadas do XGBoost.
eta	Taxa de aprendizado do XGBoost.

cp	Parâmetro de complexidade do CART: a árvore cresce só com divisões que melhoram o ajuste relativo em pelo menos cp; zero cresce a árvore máxima permitida por max_depth e min_n, depois podada.
poda	Poda do CART por custo-complexidade, escolhida pela validação cruzada do rpart com min(10, n) folds (Breiman et al. 1984): "1ep" fica com a menor árvore cujo erro de validação não passa do mínimo mais um erro-padrão; "minimo", com a de menor erro; "nenhuma" não poda.
corte	Na logística binária, a classe prevista é a segunda quando sua probabilidade é maior ou igual a corte (entre 0 e 1, exclusivos). Ignorado pelos demais modelos.
importancia	Medida de importância da floresta (ranger): "impureza" (padrão, redução de Gini/variância, enviesada para preditores com muitos valores; Strobl et al. 2007), "permutacao" (aumento do erro fora da bolsa ao permutar o preditor, Breiman 2001: erro quadrático médio na regressão, erro de Brier na classificação) ou "impureza_corrigida" (AIR, Nembrini, König & Wright 2018). Ignorado pelos demais modelos.

Value

Um models/fit: objeto de classe c("tr_ml_fit", "tr_models_fit"), que responde ao contrato da trama.models (prever, avaliar, importância). Guarda as impressões digitais das linhas de treino (treino_impressoes), com que models/predict recusa prever linhas do teste que o modelo viu.

tr_ml_forest

Floresta aleatória com ranger.

Description

Floresta aleatória com ranger.

Usage

```
tr_ml_forest(
  dados,
  resposta = "",
  preditores = "",
  tarefa = "auto",
  trees = 200L,
  mtry = 0L,
  min_n = 5L,
  max_depth = 3L,
  importancia = "impureza",
  seed = 42L
)
```

Arguments

<code>dados</code>	Tabela de treino.
<code>resposta</code>	Nome da coluna a prever.
<code>preditores</code>	Preditores separados por vírgula; em branco, usa todas as colunas numéricas, exceto o resposta.
<code>tarefa</code>	"auto", "regressao" ou "classificacao".
<code>trees</code>	Número de árvores da floresta.
<code>mtry</code>	Preditores candidatos por divisão; zero usa raiz de p.
<code>min_n</code>	Mínimo por folha no CART; tamanho mínimo do nó em FIGS e ranger.
<code>max_depth</code>	Profundidade máxima das árvores (CART, forest e XGBoost).
<code>importancia</code>	Medida de importância da floresta (ranger): "impureza" (padrão, redução de Gini/variância, enviesada para preditores com muitos valores; Strobl et al. 2007), "permutacao" (aumento do erro fora da bolsa ao permutar o preditor, Breiman 2001: erro quadrático médio na regressão, erro de Brier na classificação) ou "impureza_corrigida" (AIR, Nembrini, König & Wright 2018). Ignorado pelos demais modelos.
<code>seed</code>	Semente local; o estado aleatório da sessão é preservado.

Value

Um modelo `tr_ml_fit`.

<code>tr_ml_linear</code>	<i>Referência linear ou logística binária.</i>
---------------------------	--

Description

Referência linear ou logística binária.

Usage

```
tr_ml_linear(
  dados,
  resposta = "",
  preditores = "",
  tarefa = "auto",
  corte = 0.5,
  seed = 42L
)
```

Arguments

<code>dados</code>	Tabela de treino.
<code>resposta</code>	Nome da coluna a prever.
<code>preditores</code>	Preditores separados por vírgula; em branco, usa todas as colunas numéricas, exceto o resposta.
<code>tarefa</code>	"auto", "regressao" ou "classificacao".
<code>corte</code>	Na logística binária, a classe prevista é a segunda quando sua probabilidade é maior ou igual a corte (entre 0 e 1, exclusivos). Ignorado pelos demais modelos.
<code>seed</code>	Semente local; o estado aleatório da sessão é preservado.

Value

Um modelo `tr_ml_fit`.

<code>tr_ml_nested_cv</code>	<i>Validação cruzada aninhada</i>
------------------------------	-----------------------------------

Description

Estima o desempenho de todo o procedimento de ajuste — busca de hiperparâmetros incluída — sem reaproveitar as linhas que escolheram o vencedor (Varma & Simon 2006). Cada fold externo roda um `tr_ml_tune()` completo só no seu treino e mede o vencedor na sua validação, que a busca nunca viu.

Usage

```
tr_ml_nested_cv(
  dados,
  resposta = "",
  preditores = "",
  modelo = "cart",
  tarefa = "auto",
  metrica = "auto",
  tentativas = 10L,
  folds_externos = 5L,
  folds = 3L,
  amplitude = "conservadora",
  estrategia = "aleatoria",
  ordem = "",
  grupo = "",
  seed = 42L
)
```

Arguments

<code>dados</code>	Tabela de treino com pelo menos duas linhas.
<code>resposta</code>	Nome de uma coluna existente em dados.
<code>preditores</code>	Preditores numéricos separados por vírgula. Vazio usa todos os numéricos, exceto resposta.
<code>modelo</code>	Família com hiperparâmetros: "cart", "figs", "forest", "svm" ou "xgboost". Com "cart", cada ajuste inclui a poda 1-EP de <code>tr_ml_fit()</code> , com sua própria validação cruzada interna (até 10 ajustes extras); cp não entra na busca porque a poda já o escolhe.
<code>tarefa</code>	Uma de "auto", "regressao" ou "classificacao". "auto" interpreta resposta numérica como regressão.
<code>metrica</code>	Métrica compatível com a tarefa. "auto" usa "rmse" em regressão e "macro_f1" em classificação.
<code>tentativas</code>	Número inteiro positivo de configurações avaliadas.
<code>folds_externos</code>	Partições externas, a partir de dois.
<code>folds</code>	Partições internas de cada busca.
<code>amplitude</code>	Limites "conservadora" ou "ampla" para a busca.
<code>estrategia</code>	Como formar os folds: "aleatoria" (padrão; estratificada pela classe), "grupo" (grupos inteiros de grupo por fold), "grupo_estratificado" (grupos inteiros, distribuídos para equilibrar as classes entre os folds, como o StratifiedGroupK-Fold do scikit-learn; só classificação) ou "temporal" (origem móvel com janela crescente: os instantes de ordem formam folds + 1 blocos contíguos e cada fold valida o bloco seguinte ao treino).
<code>ordem</code>	Coluna de tempo para <code>estrategia = "temporal"</code> .
<code>grupo</code>	Coluna de grupo para <code>estrategia = "grupo"</code> . Com preditores vazio, ordem e grupo nunca entram como preditores.
<code>seed</code>	Inteiro entre zero e 2147483647. Controla folds, configurações e ajustes sem alterar o estado aleatório da sessão.

Value

Tibble com uma linha por fold externo (fold, n_treino, n_validacao, tentativa, interna = média dos folds internos do vencedor, otimista; externa = métrica na validação externa) e uma linha final fold = "media" com as médias; a estimativa honesta é externa. Com folds externos de validação de uma classe só, avisa e guarda a nota no atributo nota.

Examples

```
d <- tr_ml_example("iris_binaria")
tr_ml_nested_cv(d, resposta = "Species", tentativas = 3, folds_externos = 3, folds = 3)
```

tr_ml_residuals	<i>Gráfico de resíduos de regressão</i>
-----------------	---

Description

Gráfico de resíduos de regressão

Usage

```
tr_ml_residuals(  
  dados,  
  resposta = "",  
  predito = "previsto",  
  aspecto = "16:9",  
  tema = "padrão",  
  titulo = "",  
  rotulo_x = "",  
  rotulo_y = "",  
  legenda = "direita"  
)
```

Arguments

dados	Tabela com valores observados e previstos.
resposta	Nome da coluna observada.
predito	Nome da coluna prevista; o padrão previsto é a coluna que a models/predict escreve.
aspecto	Proporção do gráfico: "16:9", "4:3", "1:1", "3:4" ou "2:1".
tema	Nome de um tema registrado no projeto.
titulo	Título do gráfico. Vazio omite o título.
rotulo_x	Rótulo do eixo X. Vazio mantém "Previsão".
rotulo_y	Rótulo do eixo Y. Vazio mantém o rótulo de resíduo.
legenda	Posição "direita" ou "abaixo". "nenhuma" a omite.

Value

Objeto ggplot dos resíduos contra as previsões recebidas.

tr_ml_rules	Regras legíveis de CART ou FIGS
-------------	---------------------------------

Description

Regras legíveis de CART ou FIGS

Usage

```
tr_ml_rules(modelo)
```

Arguments

modelo Objeto criado por [tr_ml_fit\(\)](#); outro modelo é recusado com erro `tr_ml_error_not_fit`.

Value

Tibble com as regras exibidas pelo engine.

tr_ml_split	Divide dados em treino e teste
-------------	--------------------------------

Description

Faz uma divisão reprodutível. Para alvos categóricos, amostra cada classe nos dois subconjuntos e recusa classes singleton, que não podem ser representadas em treino e teste.

Usage

```
tr_ml_split(  
  dados,  
  resposta = "",  
  proporcao = 0.75,  
  estratificar = TRUE,  
  estrategia = "aleatoria",  
  ordem = "",  
  grupo = "",  
  seed = 42L  
)
```

Arguments

<code>dados</code>	Data frame ou tibble com as observações.
<code>resposta</code>	Nome da coluna resposta.
<code>proporcao</code>	Fração entre 0 e 1 destinada ao treino.
<code>estratificar</code>	Se TRUE, preserva cada classe categórica nos dois lados.
<code>estrategia</code>	"aleatoria" (padrão) sorteia linhas; "temporal" põe no treino as linhas mais antigas segundo ordem e no teste as posteriores, sem partir instantes empatados; "grupo" sorteia grupos inteiros de grupo (nenhum grupo nos dois lados). Nas duas últimas estratificar é ignorado.
<code>ordem</code>	Coluna de tempo (número, data ou data-hora, sem ausentes) usada por <code>estrategia = "temporal"</code> .
<code>grupo</code>	Coluna que identifica indivíduo, lote ou área, usada por <code>estrategia = "grupo"</code> ; <code>proporcao</code> passa a ser a fração dos grupos.
<code>seed</code>	Semente inteira; o estado RNG do chamador é restaurado.

Value

Lista com tibbles treino e teste, marcados com o atributo `tr_ml_origem` (`papel = "treino"/"teste"` e `divisao`, um id da divisão, mais as impressões digitais das linhas do teste). Ajustar no teste (inclusive numa tabela que junta treino e teste) e avaliar o treino passam a ser recusados. A marca não cobre juntar com o teste à direita (`tr_join(outra, teste)`), remodelar (`pivot_longer/pivot_wider`), recriar a tabela à mão, reescrever ou tirar colunas da divisão, nem divisões feitas fora do `ml/split`; aí os blocos seguem como sem marca.

tr_ml_svm

*Máquina de vetores de suporte com e1071.***Description**

Máquina de vetores de suporte com e1071.

Usage

```
tr_ml_svm(
  dados,
  resposta = "",
  preditores = "",
  tarefa = "auto",
  cost = 1,
  gamma = 0.1,
  kernel = "radial",
  seed = 42L
)
```

Arguments

dados	Tabela de treino.
resposta	Nome da coluna a prever.
preditores	Preditores separados por vírgula; em branco, usa todas as colunas numéricas, exceto o resposta.
tarefa	"auto", "regressao" ou "classificacao".
cost	Penalidade dos erros da SVM, estritamente positiva.
gamma	Escala do kernel da SVM.
kernel	Kernel linear, radial, polynomial ou sigmoid.
seed	Semente local; o estado aleatório da sessão é preservado.

Value

Um modelo tr_ml_fit.

tr_ml_tree_plot	<i>Visualizar uma árvore CART, FIGS ou de uma floresta aleatória</i>
-----------------	--

Description

Visualizar uma árvore CART, FIGS ou de uma floresta aleatória

Usage

```
tr_ml_tree_plot(
  modelo,
  arvore = 1L,
  mostrar_n = TRUE,
  mostrar_impureza = FALSE,
  casas = 3L,
  aspecto = "16:9",
  tema = "padrão",
  titulo = "",
  rotulo_x = "",
  rotulo_y = "",
  legenda = "direita"
)
```

Arguments

modelo	Modelo CART, FIGS ou floresta devolvido por <code>tr_ml_fit()</code> ; outro modelo é recusado com erro <code>tr_ml_error_not_fit</code> .
arvore	Índice positivo da árvore no FIGS ou na floresta. Ignorado pelo CART.
mostrar_n	Inclui em cada nó o número de observações que o alcançam.

mostrar_impureza	Inclui a impureza do CART ou o ganho do FIGS.
casas	Número inteiro, de zero a seis, de casas decimais nos rótulos.
aspecto	Proporção do gráfico: "16:9", "4:3", "1:1", "3:4" ou "2:1".
tema	Nome de um tema registrado no projeto.
titulo	Título do gráfico. Vazio omite o título.
rotulo_x	Rótulo do eixo X. Vazio mantém o rótulo gerado.
rotulo_y	Rótulo do eixo Y. Vazio mantém o rótulo gerado.
legenda	Posição "direita" ou "abaixo". "nenhuma" a omite.

Value

Objeto ggplot com os nós, ramos e folhas da árvore selecionada.

tr_ml_tune

Ajustar hiperparâmetros por validação cruzada

Description

O conjunto recebido é usado para validação interna; após escolher a melhor configuração, o modelo é reajustado em todas as linhas. O conjunto de teste final fica fora deste nó.

Usage

```
tr_ml_tune(
  dados,
  resposta = "",
  preditores = "",
  modelo = "cart",
  tarefa = "auto",
  metrica = "auto",
  tentativas = 20L,
  folds = 5L,
  amplitude = "conservadora",
  estrategia = "aleatoria",
  ordem = "",
  grupo = "",
  seed = 42L
)
```

Arguments

dados	Tabela de treino com pelo menos duas linhas.
resposta	Nome de uma coluna existente em dados.
preditores	Preditores numéricos separados por vírgula. Vazio usa todos os numéricos, exceto resposta.
modelo	Família com hiperparâmetros: "cart", "figs", "forest", "svm" ou "xgboost". Com "cart", cada ajuste inclui a poda 1-EP de <code>tr_ml_fit()</code> , com sua própria validação cruzada interna (até 10 ajustes extras); cp não entra na busca porque a poda já o escolhe.
tarefa	Uma de "auto", "regressao" ou "classificacao". "auto" interpreta resposta numérica como regressão.
metrica	Métrica compatível com a tarefa. "auto" usa "rmse" em regressão e "macro_f1" em classificação.
tentativas	Número inteiro positivo de configurações avaliadas.
folds	Número inteiro de partições, a partir de dois. Não pode superar o número de linhas nem o tamanho da menor classe.
amplitude	Limites "conservadora" ou "ampla" para a busca.
estrategia	Como formar os folds: "aleatoria" (padrão; estratificada pela classe), "grupo" (grupos inteiros de grupo por fold), "grupo_estratificado" (grupos inteiros, distribuídos para equilibrar as classes entre os folds, como o StratifiedGroupK-Fold do scikit-learn; só classificação) ou "temporal" (origem móvel com janela crescente: os instantes de ordem formam folds + 1 blocos contíguos e cada fold valida o bloco seguinte ao treino).
ordem	Coluna de tempo para estrategia = "temporal".
grupo	Coluna de grupo para estrategia = "grupo". Com preditores vazio, ordem e grupo nunca entram como preditores.
seed	Inteiro entre zero e 2147483647. Controla folds, configurações e ajustes sem alterar o estado aleatório da sessão.

Value

Objeto `tr_ml_tuning`: lista com o modelo vencedor reajustado (um `models/fit`), historico por tentativa e fold, índice `melhor_tentativa`, `metrica`, direção minimizar, número de folds, `seed` e nota (NULL, ou o aviso de folds de validação com uma classe só).

tr_ml_tuning_plot

*Visualizar o histórico de uma busca de hiperparâmetros***Description**

Visualizar o histórico de uma busca de hiperparâmetros

Usage

```
tr_ml_tuning_plot(
  dados,
  hiperparametro = "",
  aspecto = "16:9",
  tema = "padrão",
  titulo = "",
  rotulo_x = "",
  rotulo_y = "",
  legenda = "direita"
)
```

Arguments

<code>dados</code>	Histórico devolvido por <code>tr_ml_tune()</code> .
<code>hiperparametro</code>	Nome de uma coluna numérica do histórico. Vazio mostra a evolução das tentativas e do melhor valor acumulado.
<code>aspecto</code>	Proporção do gráfico: "16:9", "4:3", "1:1", "3:4" ou "2:1".
<code>tema</code>	Nome de um tema registrado no projeto.
<code>titulo</code>	Título do gráfico. Vazio omite o título.
<code>rotulo_x</code>	Rótulo do eixo X. Vazio mantém o rótulo gerado.
<code>rotulo_y</code>	Rótulo do eixo Y. Vazio mantém "Métrica média".
<code>legenda</code>	Posição "direita" ou "abaixo". "nenhuma" a omite.

Value

Objeto ggplot da evolução do tuning ou da relação entre um hiperparâmetro e a métrica média.

tr_ml_xgboost	<i>Árvores impulsionadas por gradiente com XGBoost.</i>
---------------	---

Description

Árvores impulsionadas por gradiente com XGBoost.

Usage

```
tr_ml_xgboost(
  dados,
  resposta = "",
  preditores = "",
  tarefa = "auto",
  nrounds = 100L,
  max_depth = 3L,
  eta = 0.1,
  seed = 42L
)
```

Arguments

<code>dados</code>	Tabela de treino.
<code>resposta</code>	Nome da coluna a prever.
<code>preditores</code>	Preditores separados por vírgula; em branco, usa todas as colunas numéricas, exceto o resposta.
<code>tarefa</code>	"auto", "regressao" ou "classificacao".
<code>nrounds</code>	Rodadas do XGBoost.
<code>max_depth</code>	Profundidade máxima das árvores (CART, forest e XGBoost).
<code>eta</code>	Taxa de aprendizado do XGBoost.
<code>seed</code>	Semente local; o estado aleatório da sessão é preservado.

Value

Um modelo `tr_ml_fit`.

<code>trama_collection</code>	<i>Coleção de aprendizado de máquina supervisionado.</i>
-------------------------------	--

Description

Carregue `trama.data`, `trama.view` e `trama.models` antes de `trama.ml`: os ajustes saem no tipo `models/fit`, e `prever`, `avaliar`, a matriz de confusão, a ROC e a importância são os blocos da `trama.models`. Motores de modelos são opcionais.

Usage

```
trama_collection()
```

Value

Uma declaração `tr_collection`.

Index

`tr_ml_cart`, [2](#)
`tr_ml_example`, [3](#)
`tr_ml_figs`, [4](#)
`tr_ml_fit`, [4](#)
`tr_ml_fit()`, [9](#), [11](#), [13](#), [15](#)
`tr_ml_forest`, [6](#)
`tr_ml_linear`, [7](#)
`tr_ml_nested_cv`, [8](#)
`tr_ml_residuals`, [10](#)
`tr_ml_rules`, [11](#)
`tr_ml_split`, [11](#)
`tr_ml_svm`, [12](#)
`tr_ml_tree_plot`, [13](#)
`tr_ml_tune`, [14](#)
`tr_ml_tune()`, [8](#), [16](#)
`tr_ml_tuning_plot`, [15](#)
`tr_ml_xgboost`, [16](#)
`trama_collection`, [17](#)