



Recipe:

recipe(.formula, .data)

#Creates a recipe.

prep() #Blends the required preprocessing transformations.

bake(new_data= NULL) #Applies a recipe to a dataset.

Standardization and Normalization:

step_log()

#Normalizes variables by applying log transformation.

step_normalize()

#Normalizes variables by forcing mean = 0, sd = 1.

step_center()

#Normalizes variables by forcing mean = 0.

step_scale()

#Normalizes variables by forcing sd = 1.

Handling Missing Data:

step_impute_median()

#Imputes missing numeric values with median.

step_impute_mean()

#Imputes missing numeric values with mean.

step_impute_mode()

#Imputes missing nominal values with mode.

Encoding Categorical Variables:

step_dummy()

#Creates dummy variables for categorical variables.

step_other()

#Collapses infrequent factors into an "other" category.

Feature Engineering:

step_mutate()

#Creates a new feature from an existing one.

step_interact()

#Creates interaction terms between features.

Tidymodels Cheatsheet

Data Filtering and Columns Selection:

step_rm(!c()) #Selects specific variables to retain in the dataset.

step_slice() #Filters rows using their position.

step_filter() #Filters rows using logical condition.

all_nominal_predictors()

#Selects all categorical predictor variables in the dataset.

all_numeric_predictors()

#Selects all numeric predictor variables in the dataset.

Linear Regression:

linear_reg()

#Specifies a linear reg model.

set_engine("lm")

#Sets the computational engine to fit the data.

set_mode()

#Defines the model's purpose (regression/classification).

Ridge & Lasso Regression:

Useful to address multicollinearity or perform variables selection.

linear_reg(penalty=0.1, mixture=)

set_engine("glmnet")

#Sets the computational engine to fit the data.

Decision Tree:

Useful to capture also non-linear relationships.

decision_tree() #Specifies a decision tree model.

set_engine("rpart")

#Sets the computational engine to fit the data.

Logistic Regression:

Predicts the probability of an outcome that can be one of two possible states.

logistic_reg() #Specifies a logistic reg model.

Gradient Boosted Models:

Automatically handles missing values and does not require scaling of data.

boost_tree() #Specifies gradient boosted model.



Random Forest:

Delivers high accuracy & it's robust against overfitting.

rand_forest() #Specifies random forest model.

Correlation Matrix:

cor(use="complete.obs")

#Creates a correlation matrix.

corrplot()

#Visualizes the correlation matrix.

Workflows:

workflow() #Combines recipe and model together.

add_recipe() #Adds your recipe to the workflow.

add_model() #Adds your model to the workflow.

fit(.data)

#Fits/runs/trains the workflow on the data.



Data Splitting:

initial_split(.data, prop=) #Defines your split and data proportion.

training() #Creates a train dataset.

testing() #Creates a test dataset.

Making Predictions:

predict(workflow, .data) #Makes predictions on test using fitted workflow.

bind_rows() #Puts all predictions metrics in one dataset.

bind_cols() #Adds predictions to the original data.



Model Evaluation & Metrics:

metrics(truth =, estimate =)

#Extracts rmse, mae and rsq.

truth #Indicates actual dependent variable values.

estimate #Indicates predicted values of dependent variables.