

R/Python Parity

The following is a list of R functions alongside the equivalent Python ones. Most methods in Python are member methods of the `H2OFrame` class. H2O does not override native Python `all` or `any` methods but these are included as member methods (e.g., `myFrame[0].any()` not `any(myFrame[0])`).

Similarly, model accessor methods are members of their respective classes.

Note: This is not a complete listing of the R or Python H2O API. Please refer to the R or Python documentation.

H2O Algorithms:

| R | Python | Function Description |
|-------------------------------|---|---|
| <code>h2o.deeplearning</code> | <code>h2o.estimators.deeplearning</code> | Creates a Deep Learning model. |
| <code>h2o.gbm</code> | <code>h2o.estimators.gbm</code> | Creates a Gradient Boosting Machine model. |
| <code>h2o.glm</code> | <code>h2o.estimators.glm</code> | Creates a Generalized Linear model. |
| <code>h2o.glmr</code> | <code>h2o.estimators.glmr</code> | Creates a Generalized Low Rank model. |
| <code>h2o.kmeans</code> | <code>h2o.estimators.kmeans</code> | Creates a K-means model. |
| <code>h2o.naiveBayes</code> | <code>h2o.estimators.naive_bayes</code> | Computes Naive Bayes probabilities on an H2O dataset. |
| <code>h2o.randomForest</code> | <code>h2o.estimators.random_forest</code> | Creates a Distributed Random Forest model. |

H2OFrame Operations:

| R | Python | Function Description |
|-------------------------------------|----------------------------|--|
| <code>as.character.H2OFrame</code> | <code>ascharacter</code> | Converts the column to characters. |
| <code>as.data.frame.H2OFrame</code> | <code>as_data_frame</code> | Returns the dataset as an R or Python object. |
| <code>as.date.H2OFrame</code> | <code>as_date</code> | Return the column with all elements converted to millis since the epoch. |
| <code>as.factor</code> | <code>asfactor</code> | Converts a column into a factor column. |
| <code>as.matrix.H2OFrame</code> | | Converts an H2OFrame to a matrix. |
| <code>as.numeric</code> | <code>asnumeric</code> | Converts factor columns to numbers (numeric columns unchanged). |
| <code>as.vector.H2OFrame</code> | | Converts an H2OFrame to a vector. |
| <code>colnames</code> | <code>col_names</code> | The column names of an H2OFrame. |
| <code>dim.H2OFrame</code> | <code>dim</code> | Returns the number of rows and columns in the H2OFrame. |
| <code>dimnames.H2OFrame</code> | <code>names</code> | Column names of an H2OFrame. |

| R | Python | Function Description |
|----------------------------------|--------------------------------|--|
| <code>h2o.anyFactor</code> | <code>anyfactor</code> | Returns whether or not the frame has factor columns. |
| <code>h2o.assign</code> | <code>assign</code> | Copies the data frame and assigns it the specified key. |
| <code>h2o.biases</code> | <code>biases</code> | (for Deep Learning) Returns the frame for the respective bias vector. |
| <code>h2o.cbind</code> | <code>cbind</code> | Takes a sequence of H2O datasets and combines them by column. |
| <code>h2o.clearLog</code> | <code>h2o.clear_log</code> | Clears all H2O R command and error response logs from the local disk for debugging. |
| <code>h2o.clusterIsUp</code> | <code>cluster_is_up</code> | Returns true if the cluster is up; false otherwise. |
| <code>h2o.clusterStatus</code> | <code>cluster_status</code> | Checks node status for the node you are connected to. |
| <code>h2o.createFrame</code> | <code>h2o.create_frame</code> | Creates an H2O data frame with real-valued, categorical, integer, and binary columns as specified. |
| <code>h2o.cut</code> | <code>cut</code> | Cuts a numeric vector into factor “buckets”. Similar to R’s <code>cut</code> method. |
| <code>h2o.day</code> | <code>day</code> | Returns a new day column from a msec-since-EPOCH column. |
| <code>h2o.dayOfWeek</code> | <code>dayOfWeek</code> | Returns a new Day-of-Week column from a msec-since-EPOCH column. |
| <code>h2o.ddply</code> | <code>ddply</code> | For each subset of an H2O data set, apply a user-specified function, then combine the results. Caution: This is an experimental feature. |
| <code>h2o.describe</code> | <code>describe</code> | Generates an in-depth description of the H2OFrame, including everything in <code>summary()</code> plus the data layout. |
| <code>h2o.downloadAllLogs</code> | <code>download_all_logs</code> | Downloads H2O log Files to the disk. |
| <code>h2o.downloadCSV</code> | <code>download_csv</code> | Downloads an H2O data set to a CSV file on the local disk. Caution: Files located on the H2O server may be very large! Make sure you have enough hard drive space to accommodate the entire file. |
| <code>h2o.entropy</code> | <code>entropy</code> | For each string, return the Shannon entropy. If the string is empty, the entropy is 0. |

| R | Python | Function Description |
|----------------------------------|-------------------------------|--|
| <code>h2o.exportFile</code> | <code>export_file</code> | Export a given H2OFrame (which can be either VA or FV) to a path on the machine this python session is currently connected to. |
| <code>h2o.filterNACols</code> | <code>filter_na_cols</code> | Filter columns that have a proportion of NAs which is $\geq \text{frac}$ |
| <code>h2o.getTimezone</code> | <code>h2o.get_timezone</code> | Returns the time zone for the H2O Cloud. |
| <code>h2o.groupBy</code> | <code>group_by</code> | Returns a new GroupBy object using this frame and the desired grouping columns. The returned groups are sorted by the natural group-by column sort. |
| <code>h2o.gsub</code> | <code>gsub</code> | Replaces all matches. Note: Changes the frame. |
| <code>h2o.head</code> | <code>head</code> | Returns the first or last rows of an H2OFrame object. Analogous to R's head call on a data.frame. |
| <code>h2o.hist</code> | <code>hist</code> | Computes a histogram over a numeric column. If <code>breaks=="FD"</code> , the MAD is used over the IQR in computing bin width. |
| <code>h2o.hit_ratio_table</code> | <code>hit_ratio_table</code> | Retrieves the Hit Ratios. |
| <code>h2o.hour</code> | <code>hour</code> | Returns a new Hour-of-Day column from a msec-since-Epoch column. |
| <code>h2o.ifelse</code> | <code>ifelse</code> | Equivalent to <code>[y if t else n for t,y,n in zip(self,yes,no)]</code> . Based on the booleans in the test vector, the output has the values of the yes and no vectors interleaved (or merged together). All Frames must have the same row count. Single column frames are broadened to match wider Frames. Scalars are allowed, and are also broadened to match wider frames. |

| R | Python | Function Description |
|--------------------------------------|------------------------------------|---|
| <code>h2o.impute</code> | <code>impute</code> | Perform in-place imputation by filling missing values with aggregates computed on the “na.rm’d” vector. Additionally, it is possible to perform imputation based on groupings of columns from within data; these columns can be passed by index or name to the <code>by</code> parameter. If a factor column is supplied, then the method must be <code>mode</code> . |
| <code>h2o.insertMissingValues</code> | <code>insert_missing_values</code> | Primarily used for testing. Randomly replaces a user-specified fraction of entries in a H2O dataset with missing values. Caution: This will modify the original dataset. Unless this is intended, this function should only be called on a subset of the original. |
| <code>h2o.interaction</code> | <code>h2o.interaction</code> | Creates an H2O frame with n-th order interaction features between categorical columns. |
| <code>h2o.is.character</code> | <code>ischaracter</code> | Returns <code>true</code> if the column is a character column, otherwise <code>false</code> . (In Python, this is the same as <code>isstring</code> .) |
| <code>h2o.is.factor</code> | <code>isfactor</code> | Returns <code>true</code> if this vector is a factor. |
| <code>h2o.is.numeric</code> | <code>isnumeric</code> | Returns <code>true</code> if the column is numeric, otherwise returns <code>false</code> . |
| <code>h2o.kfold_column</code> | <code>kfold_column</code> | Build a fold assignments column for cross-validation. This call will produce a column having the same data layout as the calling object. |
| <code>h2o.levels</code> | <code>levels</code> | Returns the factor levels for this frame and the specified column index. |
| <code>h2o.listTimezones</code> | <code>h2o.list_timezones</code> | Returns a list of all the timezones. |
| <code>h2o.loadModel</code> | <code>h2o.load_model</code> | Loads a saved H2O model from the disk. |
| <code>h2o.ls</code> | <code>ls</code> | Lists Keys on an H2O Cluster. |

| R | Python | Function Description |
|---------------------------------------|-----------------------------------|--|
| <code>h2o.lstrip</code> | <code>lstrip</code> | Strip set from the left, then return a copy of the target column with leading characters removed. The set argument is a string specifying the set of characters to be removed. If omitted, the set argument defaults to removing whitespace. |
| <code>h2o.match</code> | <code>match</code> | Creates a column of the positions of the first matches of its first argument in its second. |
| <code>h2o.mean</code> | <code>mean</code> | Returns the mean of the column. |
| <code>h2o.median</code> | <code>median</code> | Returns the median of this column. |
| <code>h2o.merge</code> | <code>merge</code> | Merge two datasets based on common column names. The two datasets must have at least one common column. |
| <code>h2o.mktime</code> | <code>mktime</code> | Computes msec since the Unix Epoch. |
| <code>h2o.month</code> | <code>month</code> | Returns a new month column from an msec-since-EPOCH column. |
| <code>h2o.nacnt</code> | <code>nacnt</code> | Returns the number of NAs per column. |
| <code>h2o.nchar</code> | <code>nchar</code> | Returns the number of characters in each string of a single-column H2OFrame. |
| <code>h2o.nlevels</code> | <code>nlevels</code> | Returns the number of factor levels for this frame and the specified column index. |
| <code>h2o.num_valid_substrings</code> | <code>num_valid_substrings</code> | Returns the count of all possible substrings ≥ 2 chars that are contained in the specified line-separated text file. |
| <code>h2o.quantile</code> | <code>quantile</code> | Computes quantiles over a given H2OFrame. |
| <code>h2o.rbind</code> | <code>rbind</code> | Combines H2O Datasets by Rows; takes a sequence of H2O data sets and combines them by rows. |
| <code>h2o.relevel</code> | <code>relevel</code> | Reorders levels of an H2O factor, similarly to standard R's <code>relevel()</code> . The levels of a factor are reordered such that the reference level is at level 0, remaining levels are moved down as needed. |

| R | Python | Function Description |
|------------------------------|---------------------------|---|
| <code>h2o.rep_len</code> | <code>rep_len</code> | Replicates the values in <code>data</code> in the H2O backend. |
| <code>h2o.round</code> | <code>round</code> | Returns the rounded values in the H2OFrame to the specified number of decimal digits. |
| <code>h2o.rstrip</code> | <code>rstrip</code> | Strip set from the right, then return a copy of the target column with leading characters removed. The set argument is a string specifying the set of characters to be removed. If omitted, the set argument defaults to removing whitespace. |
| <code>h2o.runif</code> | <code>runif</code> | Returns new H2OVec filled with doubles sampled uniformly from (0,1). |
| <code>h2o.scale</code> | <code>scale</code> | Centers and/or scales the columns of the H2OFrame. |
| <code>sd</code> | <code>sd</code> | Returns the standard deviation of column data. |
| <code>h2o.setLevel</code> | <code>set_level</code> | Sets all column values to one of the levels. |
| <code>h2o.setLevels</code> | <code>set_levels</code> | Applicable on a single categorical column. New domains must be aligned with the old domains. Note: This call does not copy the file, it changes the column in place. |
| <code>h2o.setTimezone</code> | <code>set_timezone</code> | Sets the time zone for the H2O Cloud. |
| <code>h2o.signif</code> | <code>signif</code> | Returns the rounded values in the H2OFrame to the specified number of significant digits. |
| <code>h2o.splitFrame</code> | <code>split_frame</code> | Splits a frame into distinct subsets of size determined by the given ratios. The number of subsets is always 1 more than the number of ratios given. |
| <code>h2o.strsplit</code> | <code>strsplit</code> | Splits the strings in the target column using the specified pattern. |
| <code>h2o.sub</code> | <code>sub</code> | Substitutes the first occurrence of pattern in a string with replacement. Note: Changes the frame. |

| R | Python | Function Description |
|----------------------------|------------------------|--|
| <code>h2o.substring</code> | <code>substring</code> | For each string, return a new string that is a substring of the original string. If <code>end_index</code> is not specified, then the substring extends to the end of the original string. If the <code>start_index</code> is longer than the length of the string or is greater than or equal to the <code>end_index</code> , an empty string is returned. Negative <code>start_index</code> is coerced to 0. |
| <code>h2o.summary</code> | <code>summary</code> | Summarizes the columns of an H2O data frame or subset of columns and rows using vector notation (e.g. <code>dataset[row, col]</code>). Summary includes min/mean/max/sigma and other rollup data. |
| <code>h2o.table</code> | <code>table</code> | Compute the counts of values appearing in a column, or co-occurrence counts between two columns, then returns a frame of the counts at each combination of factor levels. |
| <code>h2o.tail</code> | <code>tail</code> | Displays a digestible chunk of the H2OFrame starting from the end. Analogous to R's <code>tail</code> call on a <code>data.frame</code> . |
| <code>h2o.tolower</code> | <code>tolower</code> | Translates characters from upper to lower case for a particular column. Note: Changes the frame. |
| <code>h2o.toupper</code> | <code>toupper</code> | Translates characters from lower to upper case for a particular column. Note: Changes the frame. |
| <code>h2o.trim</code> | <code>trim</code> | (Applicable only to frames with one column) Trims the edge-spaces in a column of strings. |
| <code>h2o.unique</code> | <code>unique</code> | Extracts the unique values in a column. |
| <code>h2o.var</code> | <code>var</code> | Returns the variance or covariance matrix of the columns in this H2OFrame. |
| <code>h2o.week</code> | <code>week</code> | Converts the entries of an H2OFrame object from milliseconds to weeks of the week year (starting from 1). |

| R | Python | Function Description |
|-------------------------------|------------------------|---|
| <code>h2o.which</code> | <code>h2o.which</code> | Returns the H2OFrame of 1 column filled with 0-based indices for which the condition is True. |
| <code>h2o.year</code> | <code>year</code> | Convert the entries of an H2OFrame object from milliseconds to years, indexed starting from 1900. |
| <code>is.na</code> | <code>isna</code> | Returns a new boolean H2OVec. |
| <code>na.omit.H2OFrame</code> | <code>na_omit</code> | Removes rows with NAs from the H2OFrame. |
| <code>names.H2OFrame</code> | <code>names</code> | Column names of an H2OFrame. |
| <code>ncol.H2OFrame</code> | <code>ncol</code> | Returns the number of columns in this H2OFrame. |
| <code>nrow.H2OFrame</code> | <code>nrow</code> | Returns the number of rows in this H2OFrame. |
| <code>str.H2OFrame</code> | <code>structure</code> | Compactly display the structure of this H2OFrame. |

H2O Model Operations:

| R | Python | Function Description |
|-----------------------------|--------------------------|--|
| <code>h2o.accuracy</code> | <code>accuracy</code> | Returns the accuracy for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.aic</code> | <code>aic</code> | Retrieves the AIC for this set of metrics. |
| <code>h2o.anomaly</code> | <code>anomaly</code> | Obtain the reconstruction error for the input test_data. |
| <code>h2o.auc</code> | <code>auc</code> | Retrieves the AUC for this set of metrics. |
| <code>h2o.betweeness</code> | <code>betweeness</code> | Returns the between cluster sum of squares. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.centers</code> | <code>centers</code> | Returns the centers for the kmeans model. |
| <code>h2o.centersSTD</code> | <code>centers_std</code> | Returns the standardized centers for the kmeans model. |

| R | Python | Function Description |
|---|---|---|
| <code>h2o.centroid_stats</code> | <code>centroid_stats</code> | Returns the centroid statistics for each cluster. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.coef</code> | <code>coef</code> | Returns the coefficients for this model. |
| <code>h2o.coef_norm</code> | <code>coef_norm</code> | Returns the normalized coefficients. |
| <code>h2o.confusionMatrix</code> | <code>confusion_matrix</code> | Returns the confusion matrix for the specified metrics/thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.cross_validation_fold_assignment</code> | <code>cross_validation_fold_assignment</code> | Retrieves the cross-validation fold assignment for all rows in the training data. |
| <code>h2o.cross_validation_holdout_predictions</code> | <code>cross_validation_holdout_predictions</code> | Retrieves the (out-of-sample) holdout predictions of all cross-validation models on the training data. This is equivalent to summing up all H2OFrames returned by <code>cross_validation_predictions</code> . |
| <code>h2o.cross_validation_models</code> | <code>cross_validation_models</code> | Retrieves a list of cross-validation models. |
| <code>h2o.cross_validation_predictions</code> | <code>cross_validation_predictions</code> | Retrieves the (out-of-sample) holdout predictions of all cross-validation models on their holdout data. Note that the predictions are expanded to the full number of rows of the training data, with 0 fill-in. |
| <code>h2o.deepfeatures</code> | <code>deepfeatures</code> | Returns hidden layer details. |
| <code>h2o.download_pojo</code> | <code>download_pojo</code> | Downloads the POJO for this model to the specified path directory - do not use a trailing slash. If path is "", then dump to screen. |

| R | Python | Function Description |
|---|---|--|
| <code>h2o.F0point5</code> | <code>F0point5</code> | Returns the F0.5 for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.F1</code> | <code>F1</code> | Returns the F1 for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.F2</code> | <code>F2</code> | Returns the F2 for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.find_threshold_by_max_metric</code> | <code>find_threshold_by_max_metric</code> | If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.fnr</code> | <code>fnr</code> | Returns the False Negative Rates for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.fpr</code> | <code>fpr</code> | Returns the False Positive Rates for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |

| R | Python | Function Description |
|---|---------------------------------------|--|
| <code>h2o.gainsLift</code> | <code>gains_lift</code> | Get the Gains/Lift table for the specified metrics. If all are False (default), then return the training metric Gains/Lift table. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.getGLMFullRegularizationPath</code> | <code>getGLMRegularizationPath</code> | Extract full regularization path explored during lambda search from glm model. |
| <code>h2o.getGrid</code> | <code>get_grid</code> | Get a grid object from H2O distributed K/V store. |
| <code>h2o.giniCoef</code> | <code>giniCoef</code> | Retrieves the Gini coefficient for this set of metrics. |
| <code>h2o.logloss</code> | <code>logloss</code> | Retrieves the log loss for this set of metrics. |
| <code>h2o.makeGLMModel</code> | <code>makeGLMModel</code> | allows setting betas of an existing GLM model. |
| <code>h2o.maxPerClassError</code> | <code>max_per_class_error</code> | Returns the max per-class error for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.mcc</code> | <code>mcc</code> | Returns the Matthews correlation coefficient (MCC) for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.mean_residual_deviance</code> | <code>mean_residual_deviance</code> | Returns the mean residual deviance for a set of metrics. |
| <code>h2o.metric</code> | <code>metric</code> | Returns the metric value for a set of thresholds. If all are False (default), then return the training metric value. If more than one option is set to True, then return a dictionary of metrics where the keys are train , valid , and xval . |
| <code>h2o.mse</code> | <code>mse</code> | Retrieves the MSE for this set of metrics. |
| <code>h2o.null_deviance</code> | <code>null_deviance</code> | Returns the null deviance if the model has residual deviance, or None if no null deviance. |

| R | Python | Function Description |
|------------------------------------|--|---|
| <code>h2o.null_dof</code> | <code>null_degrees_of_freedom</code> | Returns the null degrees of freedom if the model has residual deviance, or None if no null degrees of freedom. |
| <code>h2o.num_iterations</code> | <code>num_iterations</code> | Returns the number of iterations required for convergence or to reach max iterations. |
| <code>h2o.performance</code> | <code>model_performance</code> | Generates model metrics for this model on <code>test_data</code> . |
| <code>h2o.predict</code> | <code>predict</code> | Predicts on a dataset. |
| <code>h2o.precision</code> | <code>precision</code> | Returns the precision for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are <code>train</code> , <code>valid</code> , and <code>xval</code> . |
| <code>h2o.proj_archetypes</code> | <code>proj_archetypes</code> | Project each archetype in an H2O GLRM model into the corresponding feature space from the H2O training frame. |
| <code>h2o.r2</code> | <code>r2</code> | Retrieves the R^2 coefficient for this set of metrics. |
| <code>h2o.reconstruct</code> | <code>reconstruct</code> | Reconstruct the training data from the GLRM model and impute all missing values. |
| <code>h2o.residual_deviance</code> | <code>residual_deviance</code> | Returns the residual deviance if the model has residual deviance, or None if no residual deviance. |
| <code>h2o.residual_dof</code> | <code>residual_degrees_of_freedom</code> | Returns the residual degrees of freedom if the model has residual deviance, or None if no residual degrees of freedom. |
| <code>h2o.saveModel</code> | <code>save_model</code> | Saves an H2O model object to the disk. |
| <code>h2o.scoreHistory</code> | <code>scoring_history</code> | Retrieves model score history. |
| <code>show</code> | <code>show</code> | Returns the rounded values in the H2OFrame to the specified number of significant digits. |
| <code>h2o.size</code> | <code>size</code> | Returns the sizes of each cluster. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are <code>train</code> , <code>valid</code> , and <code>xval</code> . |
| <code>summary</code> | <code>summary</code> | Generates a summary of the frame on a per-Vec basis. |

| R | Python | Function Description |
|-------------------------------|---------------------------|--|
| <code>h2o.tnr</code> | <code>tnr</code> | Returns the True Negative Rate for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are <code>train</code> , <code>valid</code> , and <code>xval</code> . |
| <code>h2o.totss</code> | <code>totss</code> | Returns the total sum of squares to grand mean. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are <code>train</code> , <code>valid</code> , and <code>xval</code> . |
| <code>h2o.tot_withinss</code> | <code>tot_withinss</code> | Returns the total within cluster sum of squares. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are <code>train</code> , <code>valid</code> , and <code>xval</code> . |
| <code>h2o.tpr</code> | <code>tpr</code> | Returns the True Positive Rate for a set of thresholds. If all are False (default), then return the training metric value. If more than one options is set to True, then return a dictionary of metrics where the keys are <code>train</code> , <code>valid</code> , and <code>xval</code> . |
| <code>h2o.varimp</code> | <code>varimp</code> | Prettyprints the variable importances or returns them in a list ordered from most important to least important. Each entry in the list is a 4-tuple of (variable, relative_importance, scaled_importance, percentage). |
| <code>h2o.weights</code> | <code>weights</code> | Returns the frame for the respective weight matrix. |

| R | Python | Function Description |
|--|---|--|
| <code>h2o.withinss</code> | <code>withinss</code> | Returns the within cluster sum of squares for each cluster. If all are False (default), then returns the training metric value. If more than one options is set to True, then returns a dictionary of metrics where the keys are <code>train</code> , <code>valid</code> , and <code>xval</code> . |
| <code>plot.H2OModel</code> | <code>plot</code> | Plots training set scoring history (and validation set if available) for an H2O Model. |
| <code>plot.H2OTabulate</code> | | Plots the simple co-occurrence based tabulation of X vs Y as a heatmap, where X and Y are two Vectors in a given dataset |
| <code>predict.H2OModel</code> | <code>predict</code> | Predict on a dataset. |
| <code>predict_leaf_node_assignment.H2OModel</code> | <code>predict_leaf_node_assignment</code> | Predict on a dataset and return the leaf node assignment (only for tree-based models). |
| <code>print.H2OTable</code> | | Returns a truncated view of the table if there are more than 20 rows. |
| <code>summary,H2OGrid-method</code> | <code>summary</code> | Format the grid object in a user-friendly way. |
| <code>summary,H2OModel-method</code> | <code>summary</code> | Return a detailed summary of the model. |

Other Methods:

| R | Python | Function Description |
|------------------------------------|--------------------------------|---|
| <code>h2o.clusterInfo</code> | <code>cluster_info</code> | Performs node status check for connected node. |
| <code>h2o.getFrame</code> | <code>get_frame</code> | Obtains a handle to the frame in H2O with the <code>frame_id</code> key. |
| <code>h2o.getModel</code> | <code>get_model</code> | Returns the specified model. |
| <code>h2o.grid</code> | <code>H2OGridSearch</code> | Provides a set of functions to launch a grid search of a hyper-parameter space for a model get its results. |
| <code>h2o.importFile</code> | <code>import_file</code> | Imports files into an H2O cloud. The default behavior is to pass-through to the parse phase automatically. |
| <code>h2o.import_sql_select</code> | <code>import_sql_select</code> | Imports the SQL table that is the result of the specified SQL query to H2OFrame in memory. Currently supported SQL databases are MySQL, PostgreSQL, MariaDB, and Netezza. |

| R | Python | Function Description |
|---|--|---|
| <code>h2o.import_sql_table</code> | <code>import_sql_table</code> | Import SQL table to H2OFrame in memory. Assumes that the SQL table is not being updated and is stable. Currently supported SQL databases are MySQL, PostgreSQL, and MariaDB. |
| <code>h2o.init</code> | <code>h2o.init</code> | Initiates an H2O connection to the specified IP address and port. |
| <code>h2o.logAndEcho</code> | <code>log_and_echo</code> | Sends a message to H2O for logging and/or debugging purposes. |
| <code>h2o.networkTest</code> | <code>network_test</code> | View network speed with various file sizes. |
| <code>h2o.no_progress</code> | <code>no_progress</code> | Disable the progress bar from flushing to stdout. The completed progress bar is printed when a job is complete so as to demarcate a log file. |
| <code>h2o.openLog</code> | <code>open_log</code> | Opens existing logs of H2O R POST commands and error responses on the local disk. |
| <code>h2o.parseRaw</code> | <code>parse_raw</code> | Used in conjunction with <code>import_file</code> and <code>parse_setup</code> in order to make alterations before parsing. |
| <code>h2o.parseSetup</code> | <code>parse_setup</code> | During parse setup, the H2O cluster will make several guesses about the attributes of the data. This method allows a user to perform corrective measures by updating the returning dictionary from this method. This dictionary is then fed into <code>parse_raw</code> to produce the H2OFrame instance. |
| <code>h2o.removeAll</code> <code>h2o.rm</code> | <code>remove_all</code> <code>remove</code> | Removes all objects from H2O. Removes the specified object from H2O. This is a “hard” delete of the object and removes all subparts. |
| <code>h2o.show_progress</code> | <code>show_progress</code> | Enables the progress bar. (Progress bar is enabled by default). |
| <code>h2o.shutdown</code> | <code>shutdown</code> | Shuts down the specified instance. All data will be lost. |
| <code>h2o.startLogging</code> | <code>h2o.start_logging</code> | Starts logging H2O R POST commands and error responses to the local disk. |

| R | Python | Function Description |
|------------------------------|-------------------------------|--|
| <code>h2o.stopLogging</code> | <code>h2o.stop_logging</code> | Stops logging of H2O R POST commands and error responses to the local disk. |
| <code>h2o.uploadFile</code> | <code>upload_file</code> | Uploads a dataset at the path given from the local machine to the H2O cluster. |

Ops Group

This group includes:

- **Arith**, for performing arithmetic on numeric or complex vectors
- **Compare**, for comparing values
- **Logic**, for logical operations

| R | Python | Type |
|--------------------|---------------------|---------|
| <code>+</code> | <code>+</code> | Arith |
| <code>-</code> | <code>-</code> | Arith |
| <code>*</code> | <code>*</code> | Arith |
| <code>^</code> | <code>^</code> | Arith |
| <code>/</code> | <code>/</code> | Arith |
| <code>%%</code> | <code>mod</code> | Arith |
| <code>%/%</code> | <code>intDiv</code> | Arith |
| <code>==</code> | <code>==</code> | Compare |
| <code>!=</code> | <code>!=</code> | Compare |
| <code><</code> | <code><</code> | Compare |
| <code>></code> | <code>></code> | Compare |
| <code><=</code> | <code><=</code> | Compare |
| <code>>=</code> | <code>>=</code> | Compare |
| <code>&</code> | <code>&</code> | Logic |
| <code> </code> | <code> </code> | Logic |
| <code>!</code> | <code>!</code> | Logic |

Math Group

This group includes:

- **Miscellaneous**, which contains the absolute value and square root functions
- **Rounding**, which allows rounding of numbers
- **Logarithms/Exponentials**, which compute logarithmic and exponential functions
- **Trigonometric**, for trigonometric functions
- **Hyperbolic**, for hyperbolic functions
- **Sign**, which returns a vector with the signs of the corresponding elements of x (does not work for complex vectors)
- **Special**, which contains gamma functions
- **Cumulative**, which returns the cumulative sums, products, minima, or maxima

| R | Python | Type |
|--------------------|--------------------|---------------|
| <code>abs</code> | <code>abs</code> | Miscellaneous |
| <code>sqrt</code> | <code>sqrt</code> | Miscellaneous |
| <code>floor</code> | <code>floor</code> | Rounding |

| R | Python | Type |
|----------|----------|----------------------------|
| ceiling | ceil | Rounding |
| trunc | trunc | Rounding |
| exp | exp | Log/Exp |
| expm1 | expm1 | Log/Exp |
| log | log | Log/Exp |
| log10 | log10 | Log/Exp |
| log2 | log2 | Log/Exp |
| log1p | log1p | Log/Exp |
| cos | cos | Trigonometric |
| sin | sin | Trigonometric |
| tan | tan | Trigonometric |
| acos | acos | Trigonometric |
| asin | asin | Trigonometric |
| atan | atan | Trigonometric |
| cospi | cospi | Trigonometric two-argument |
| sinpi | sinpi | Trigonometric two-argument |
| tanpi | tanpi | Trigonometric two-argument |
| cosh | cosh | Hyperbolic |
| sinh | sinh | Hyperbolic |
| tanh | tanh | Hyperbolic |
| acosh | acosh | Hyperbolic |
| asinh | asinh | Hyperbolic |
| atanh | atanh | Hyperbolic |
| sign | sign | Sign |
| round | round | Sign |
| signif | signif | Sign |
| lgamma | lgamma | Special |
| gamma | gamma | Special |
| digamma | digamma | Special |
| trigamma | trigamma | Special |
| cumsum | cumsum | Cumulative |
| cumprod | cumprod | Cumulative |
| cummax | cummax | Cumulative |
| cummin | cummin | Cumulative |

Summary Group This group includes:

- **Maxima/Minima**, which returns the maxima and minima
- **Range**, which returns the range of a column
- **Product**, which returns the product
- **Sum**, which returns the sum
- **All**, which tells the user if all values are true
- **Any**, which tells the user if any values are true

| R | Python |
|-------|--------|
| max | max |
| min | min |
| range | |
| prod | prod |
| sum | sum |
| all | all |

| R | Python |
|----------|---------------|
| any | any |

Non Group Generic This group includes:

- **Extract/Replace**, for extracting or replacing part of an object
- **Matrix Multiplication**, for multiplying two matrices
- **Value Matching**, for returning matching vectors

| R | Python | Type |
|----------|---------------|-----------------------|
| [| | Extract/Replace |
| [[| | Extract/Replace |
| [[<- | | Extract/Replace |
| [<- | | Extract/Replace |
| \$<- | | Extract/Replace |
| %% | // | Matrix Multiplication |
| %x% | mult | Matrix Multiplication |
| %in% | in | Value Matching |