**Week 08: R to Python**

|  |  |  |  |
| --- | --- | --- | --- |
| **R** | **Python** | **Use** | **Difference** |
| mice::md.pattern(df) | df.isna().sum() | Show location on NA’s in a dataframe by column | R function gives a visual representation while python just gives column names and how many NA’s they contain  [Python](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.isna.html)  [R](https://www.rdocumentation.org/packages/mice/versions/3.14.0/topics/md.pattern) |
| fct\_collapse() | .replace() | Collapse a specific variable into fewer levels/classes | Pretty much the same  [Python](https://www.w3schools.com/python/ref_string_replace.asp)  [R](https://forcats.tidyverse.org/reference/fct_collapse.html) |
| Not needed | OrdinalEncoder() | Turn categorical data into numeric data, only use if data has a natural order | Still need to us fit\_transform to change data  [Python](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html) |
| Index splitting | df.drop() | Drop columns or rows from dataframe | Pandas function in python, built into R structure  [Python](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html)  [R](https://rspatial.org/intr/4-indexing.html) |
| trainControl(method= ‘repeatedcv’) | RepeatedStratifiedKFold() | Repeats Stratified K-Fold n times with different randomization in each repetition | Basically the same  [Python](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RepeatedStratifiedKFold.html)  [R](https://www.rdocumentation.org/packages/caret/versions/6.0-92/topics/trainControl) |
| train(x=features,  y=target, method="rpart2", trControl=fitControl, tuneGrid=tree.grid, metric="ROC") | 1)DecisionTreeClassifier()  2)GridSearchCV()  3).fit(X\_train,y\_train) | 1)Creates a classifier tree  2)Sets hyperparameter search  3)Trains model | R train() will conduct it’s hyperparameter search and model fitting all in one, where the python function needs several functions as seen to the left  [Python1](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html),[2](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html),[3](https://scikit-learn.org/stable/tutorial/basic/tutorial.html)  [R](https://www.rdocumentation.org/packages/caret/versions/4.47/topics/train) |
| NA | model.best\_estimator\_ | Gives best estimator from fitted model | R train() will give this automatically when the above function is run and printed  [Python](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) |
| model$results | model.cv\_results\_ | Get mean testing scores and standard deviation from model | In python the results are in a dictionary, and you must extract them using keys, simply use indexing in R  [Python](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) |
| rpart.plot() | export\_graphviz() | Print the decision tree from a desired model | Similar, in python you must pass feature names and classification labels  [Python](https://scikit-learn.org/stable/modules/generated/sklearn.tree.export_graphviz.html)  [R](https://www.rdocumentation.org/packages/rpart.plot/versions/3.1.0/topics/rpart.plot) |
| varImp(model) | estimator.feature\_importances\_ | Get the feature importance’s of the best decision tree created | In R you pass the model, in python you must pass the best estimator directly  [Python](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html)  [R](https://www.rdocumentation.org/packages/caret/versions/6.0-92/topics/varImp) |
| confusionMatrix() | metrics.ConfusionMatrixDisplay.from\_estimator | Creates a confusion matrix and displays it | Very similar, python has a more user-friendly display, but r gives more information along with it. Python you pass estimator and data, R you must pass already predicted data  [Python](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.ConfusionMatrixDisplay.html#sklearn.metrics.ConfusionMatrixDisplay.from_estimator)  [R](https://www.rdocumentation.org/packages/caret/versions/6.0-92/topics/confusionMatrix) |
| gather() | pd.melt() | Creates a vertical dataframe | Similar  [Python](https://pandas.pydata.org/docs/reference/api/pandas.melt.html)  [R](https://www.statology.org/gather-function-in-r/) |
| rpart( method = "class", parms = list(split = "gini")  control = rpart.control(cp=.001)) | Model. cost\_complexity\_pruning\_path | Compute the pruning path of decision tree model using Cost complexity pruning | Similar, but rpart has more information. cost complexity pruning in python simply gives ccp\_alphas and impurities which then one can work with and create models form manually.  [Python](https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html)  [R](https://www.rdocumentation.org/packages/rpart/versions/4.1.16/topics/rpart) |
| N/A | .score | Get accuracy score of a model based on testing or tuning data. | Simply pass feature and target data  [Python](https://scikit-learn.org/stable/modules/model_evaluation.html) |