Stata Conference 2021
5th & 6th August

Machine Learning using Stata/Python

Giovanni Cerulli

Consiglio Nazionale delle Ricerche

IRCrES

ISTITUTO di RICERCA sulla CRESCITA ECONOMICA SOSTENIBILE
INSTITUTE of RESEARCH on ECONOMIC SUSTAINABLE GROWTH

What is Machine Learning ?

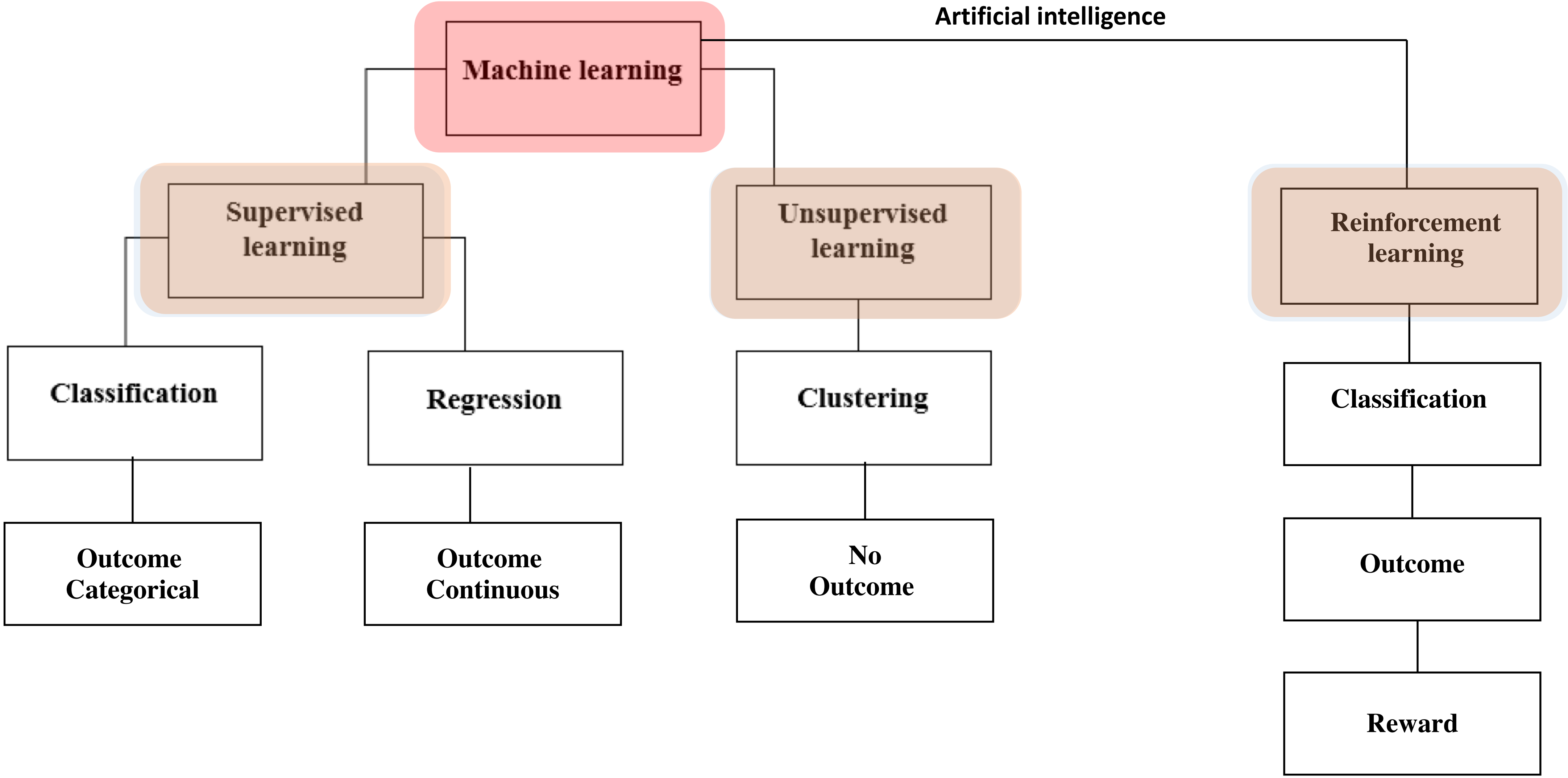
Machine Learning

A relatively new approach to **data analytics**, which places itself in the intersection between **statistics**, **computer science**, and **artificial intelligence**

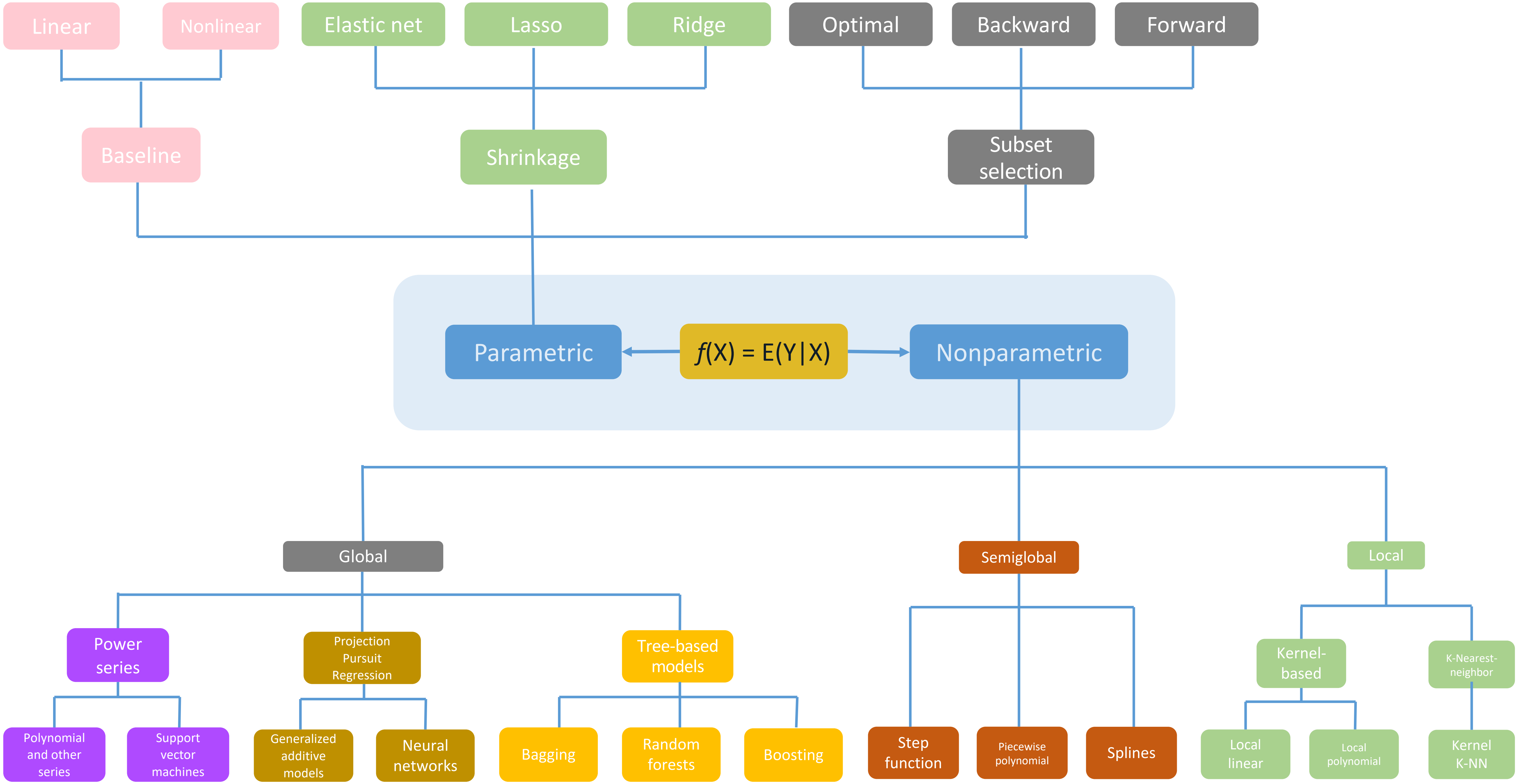
ML objective

Turning **information** into **knowledge** and **value** by **“letting the data speak”**

Supervised, Unsupervised, Reinforcement Learning



Supervised Machine Learning Methods



Linear

Nonlinear

Elastic net

Lasso

Ridge

Optimal

Backward

Forward

Baseline

Shrinkage

Subset selection

Parametric

$f(X) = E(Y|X)$

Nonparametric

Global

Semiglobal

Local

Power series

Projection Pursuit Regression

Tree-based models

Kernel-based

K-Nearest-neighbor

Polynomial and other series

Support vector machines

Generalized additive models

Neural networks

Bagging

Random forests

Boosting

Step function

Piecewise polynomial

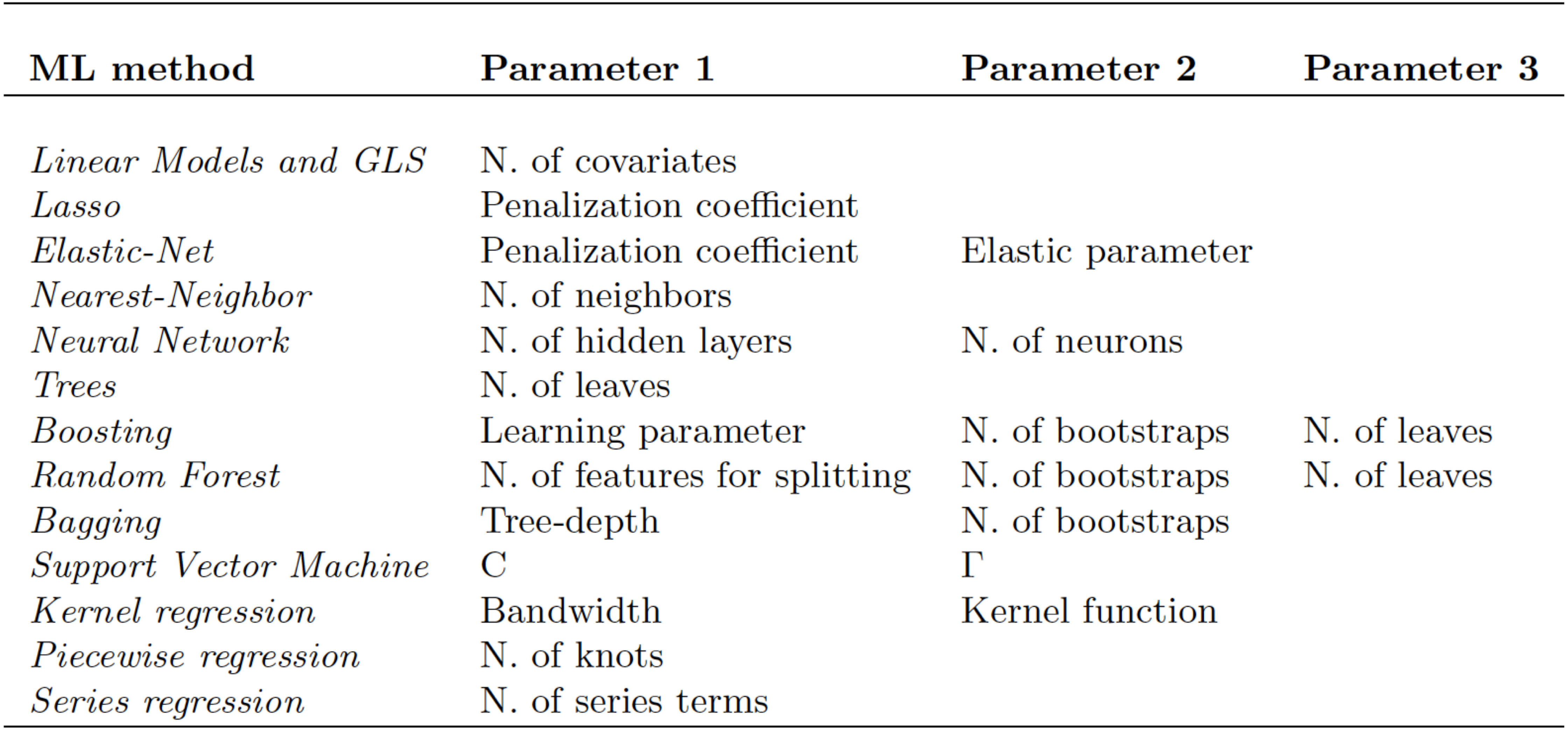
Splines

Local linear

Local polynomial

Kernel K-NN

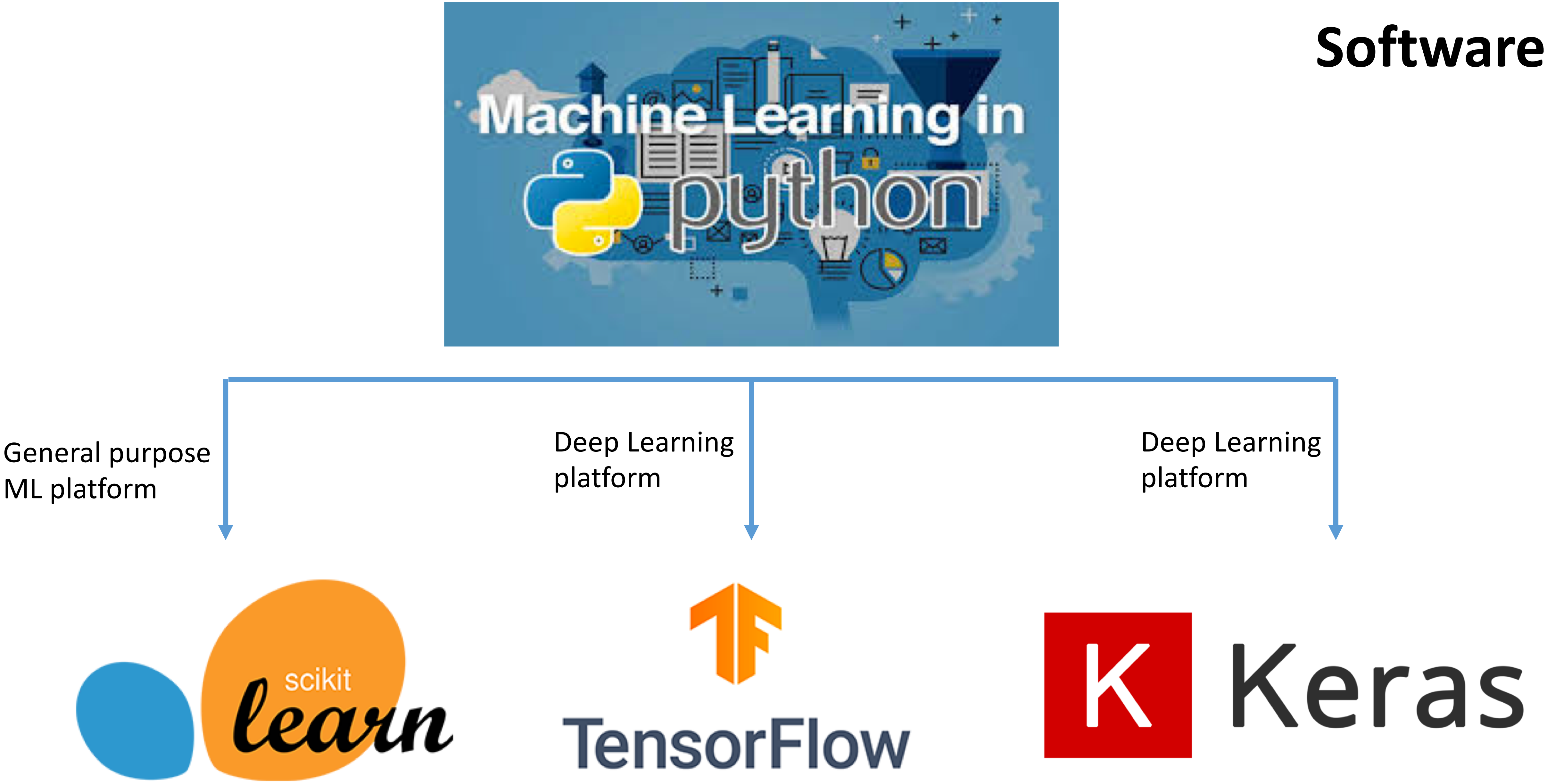
Hyper-parameter tuning



ML method	**Parameter 1**	**Parameter 2**	**Parameter 3**
Linear Models and GLS	N. of covariates		
Lasso	Penalization coefficient		
Elastic-Net	Penalization coefficient	Elastic parameter	
Nearest-Neighbor	N. of neighbors		
Neural Network	N. of hidden layers	N. of neurons	
Trees	N. of leaves		
Boosting	Learning parameter	N. of bootstraps	N. of leaves
Random Forest	N. of features for splitting	N. of bootstraps	N. of leaves
Bagging	Tree-depth	N. of bootstraps	
Support Vector Machine	C	Γ	
Kernel regression	Bandwidth	Kernel function	
Piecewise regression	N. of knots		
Series regression	N. of series terms		

Software for ML

Software

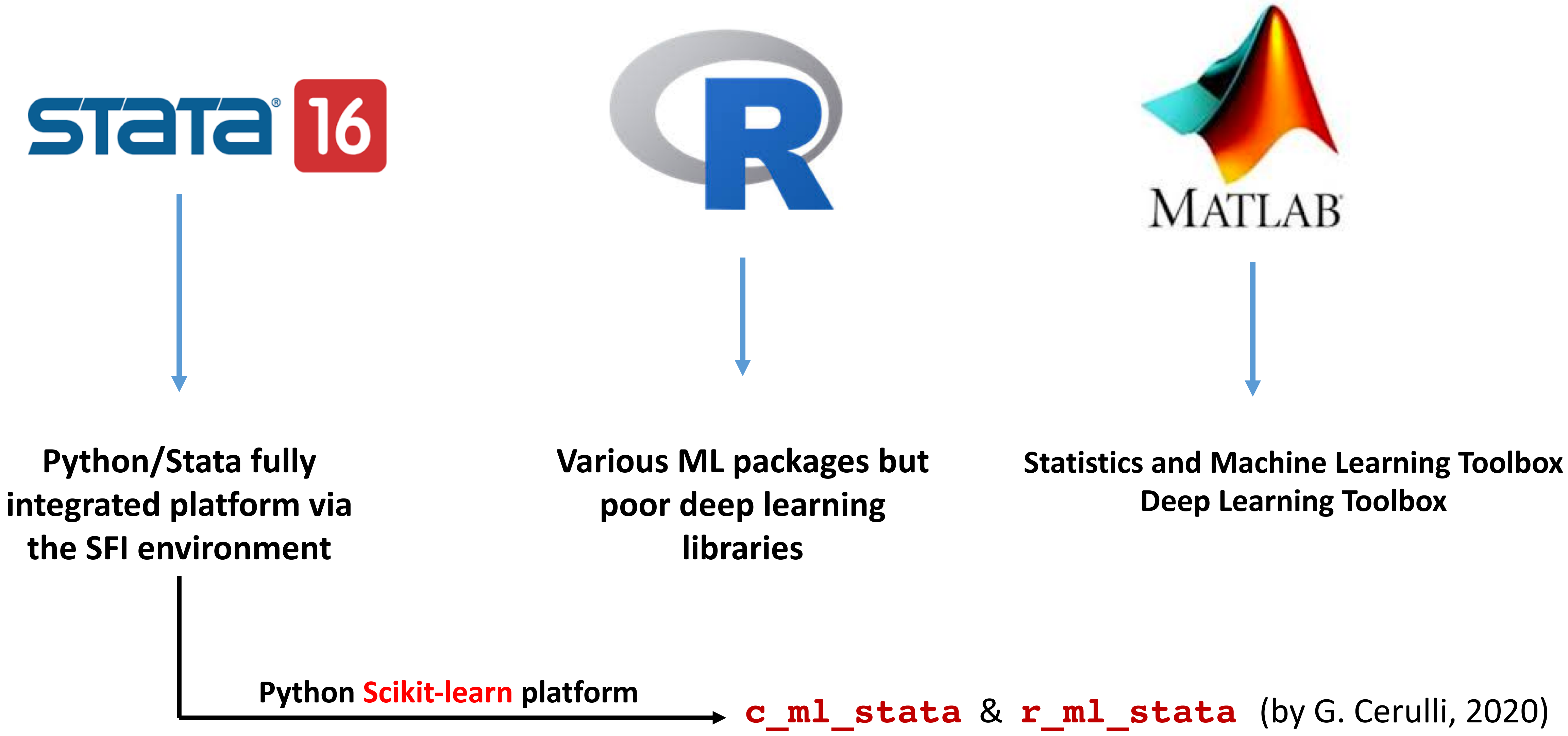


General purpose ML platform

Deep Learning platform

Deep Learning platform

Software



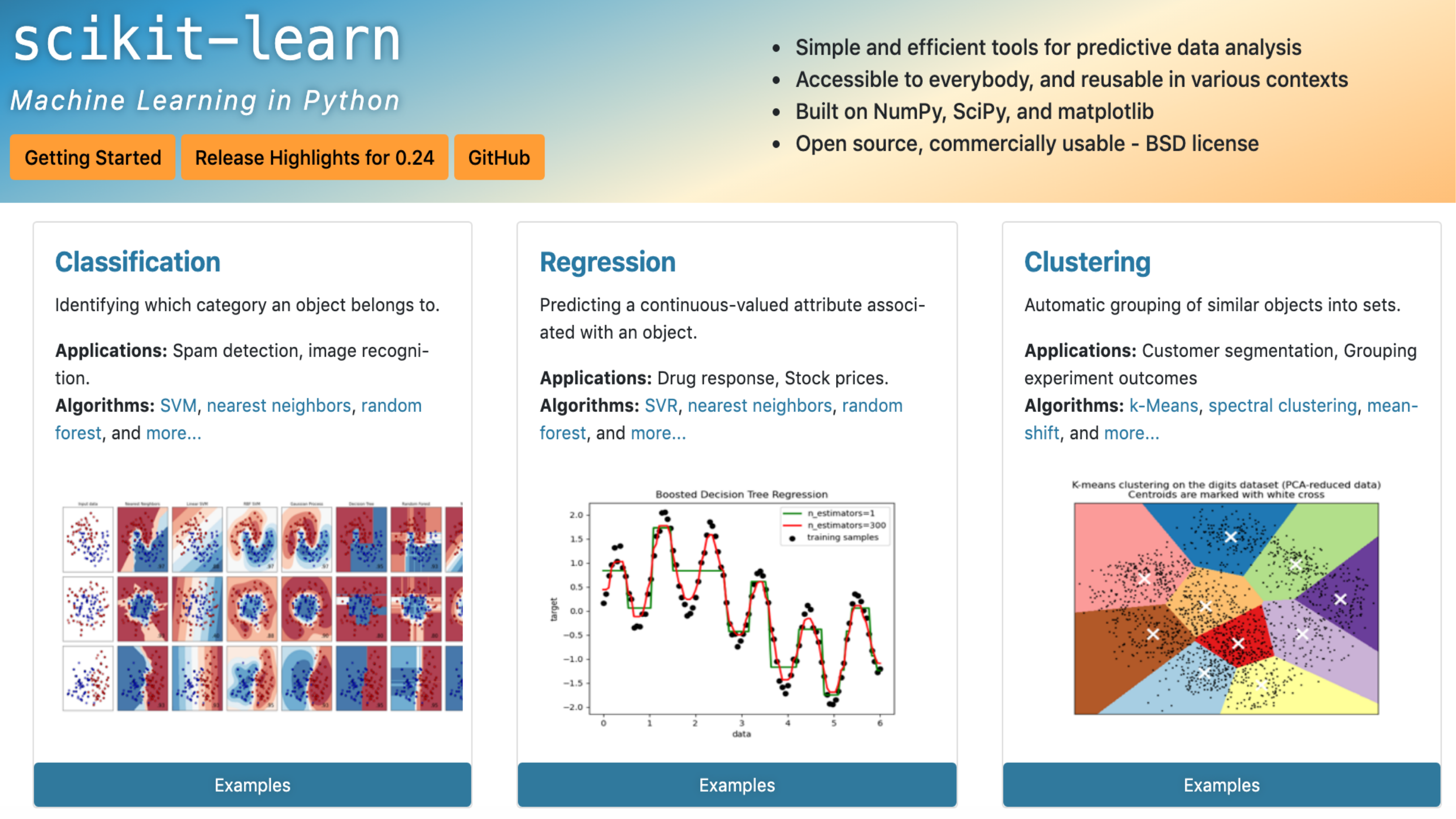
Python/Stata fully integrated platform via the SFI environment

Various ML packages but poor deep learning libraries

Statistics and Machine Learning Toolbox
Deep Learning Toolbox

Python Scikit-learn platform

`c_ml_stata` & `r_ml_stata` (by G. Cerulli, 2020)



scikit-learn

Machine Learning in Python

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross

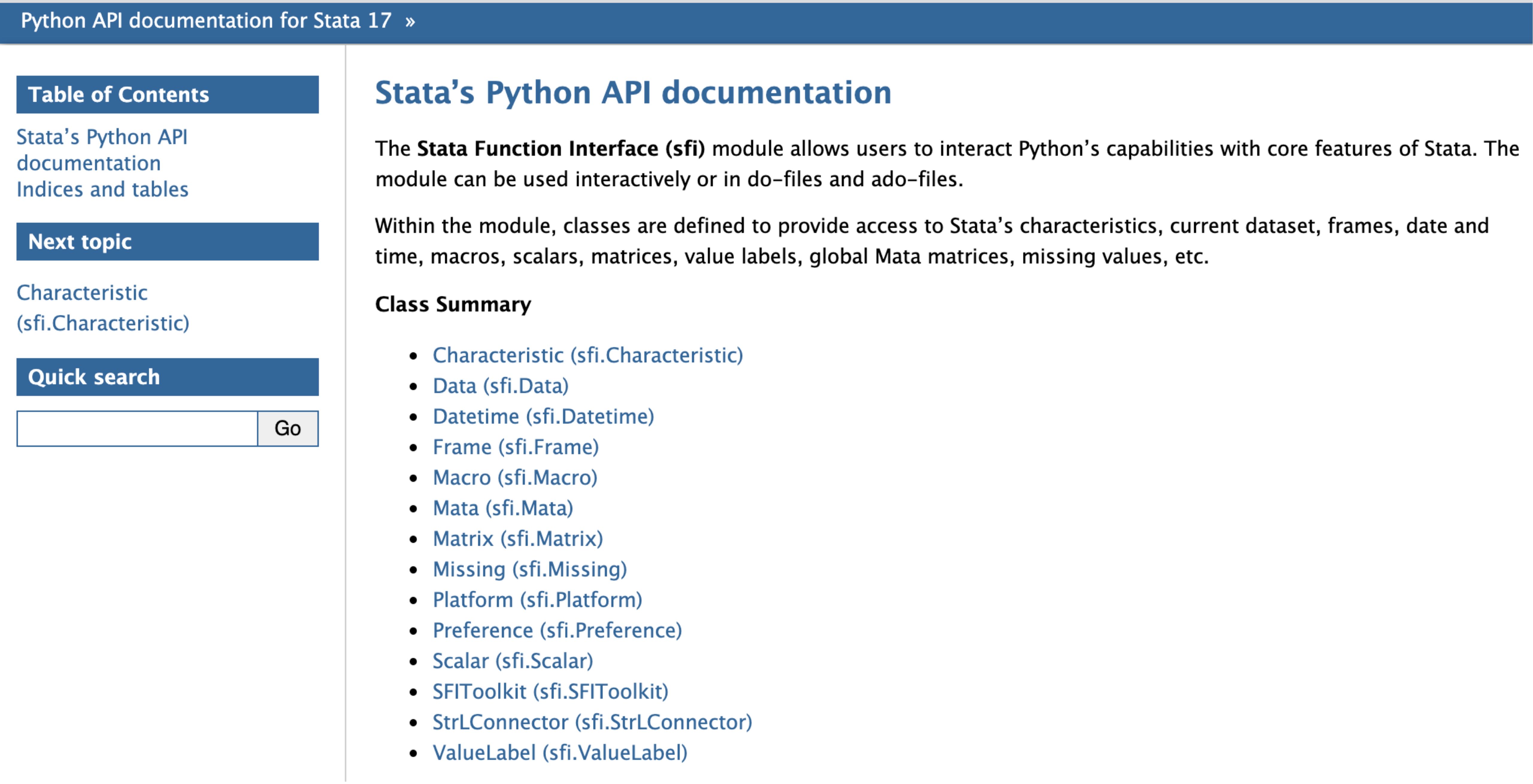


Table of Contents

Next topic

Characteristic (sfi.Characteristic)

Quick search

Stata’s Python API documentation

The **Stata Function Interface (sfi)** module allows users to interact Python’s capabilities with core features of Stata. The module can be used interactively or in do-files and ado-files.

Within the module, classes are defined to provide access to Stata’s characteristics, current dataset, frames, date and time, macros, scalars, matrices, value labels, global Mata matrices, missing values, etc.

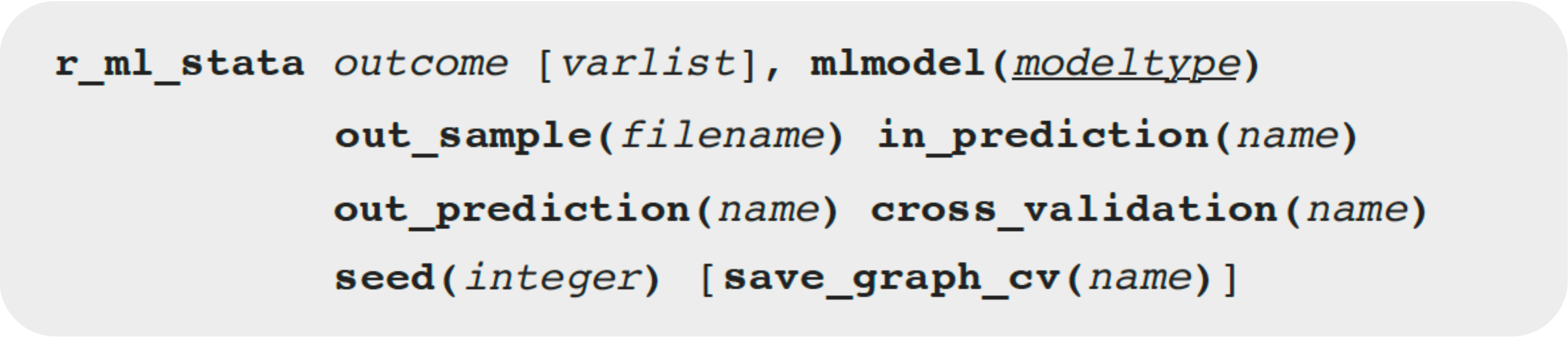
Class Summary

- Characteristic (sfi.Characteristic)
- Data (sfi.Data)
- Datetime (sfi.Datetime)
- Frame (sfi.Frame)
- Macro (sfi.Macro)
- Mata (sfi.Mata)
- Matrix (sfi.Matrix)
- Missing (sfi.Missing)
- Platform (sfi.Platform)
- Preference (sfi.Preference)
- Scalar (sfi.Scalar)
- SFIToolkit (sfi.SFIToolkit)
- StrLConnector (sfi.StrLConnector)
- ValueLabel (sfi.ValueLabel)

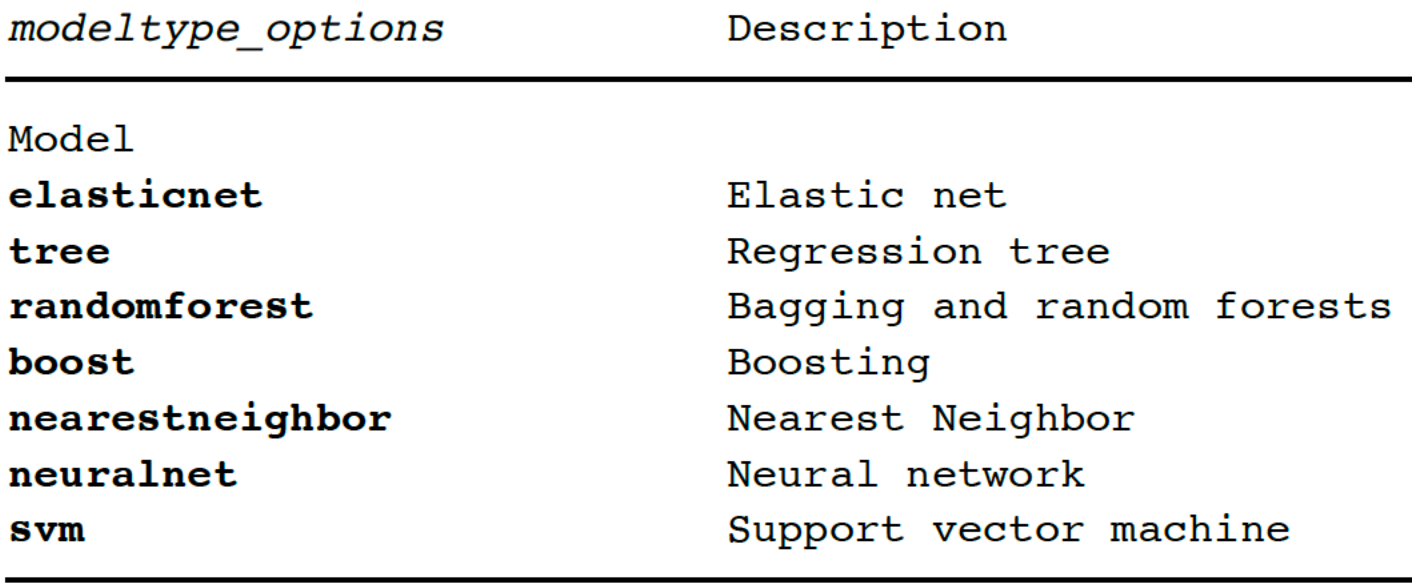
ML regression and classification with

`r_ml_stata` & `c_ml_stata`

Stata command r_ml_stata



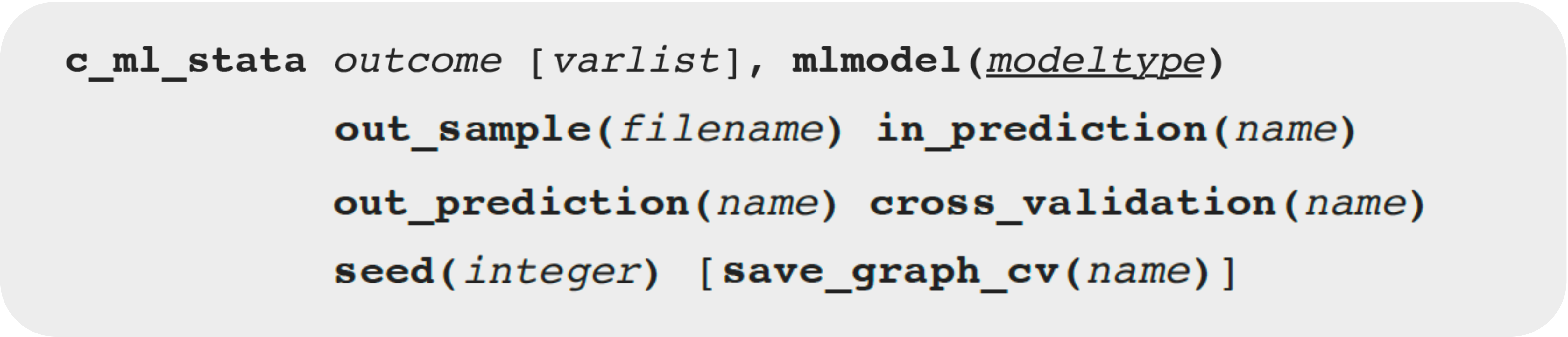
```
r_ml_stata outcome [varlist], mlmodel(modeltype)
           out_sample(filename) in_prediction(name)
           out_prediction(name) cross_validation(name)
           seed(integer) [save_graph_cv(name)]
```



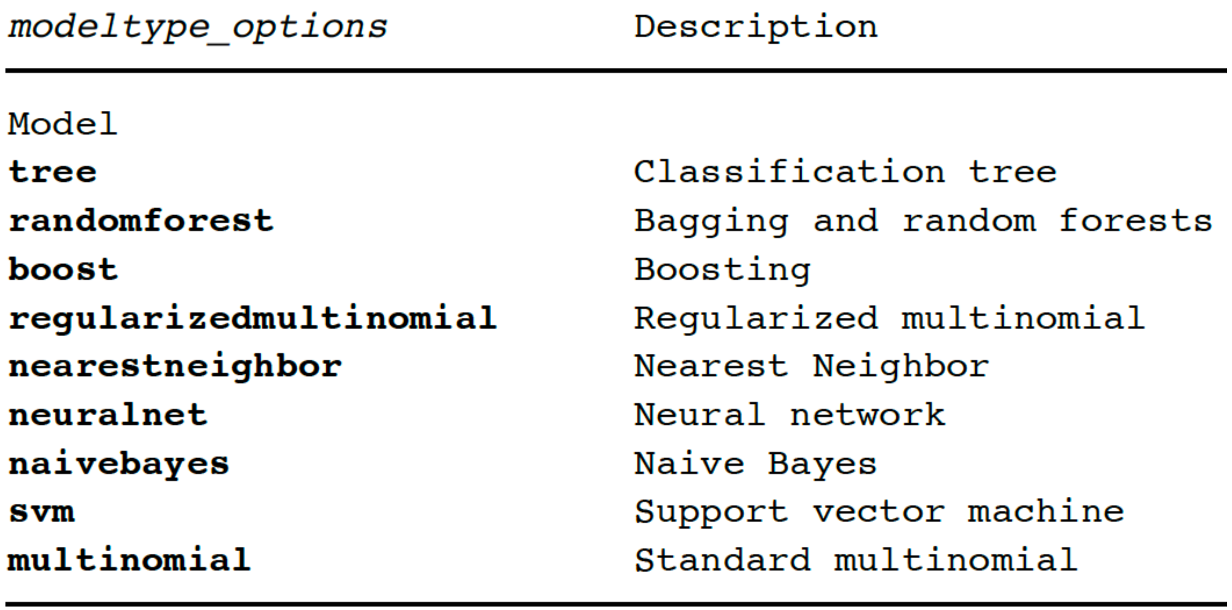
modeltype_options	Description
Model	
elasticnet	Elastic net
tree	Regression tree
randomforest	Bagging and random forests
boost	Boosting
nearestneighbor	Nearest Neighbor
neuralnet	Neural network
svm	Support vector machine

Regression

Stata command c_ml_stata



```
c_ml_stata outcome [varlist], mlmodel(modeltype)
           out_sample(filename) in_prediction(name)
           out_prediction(name) cross_validation(name)
           seed(integer) [save_graph_cv(name)]
```

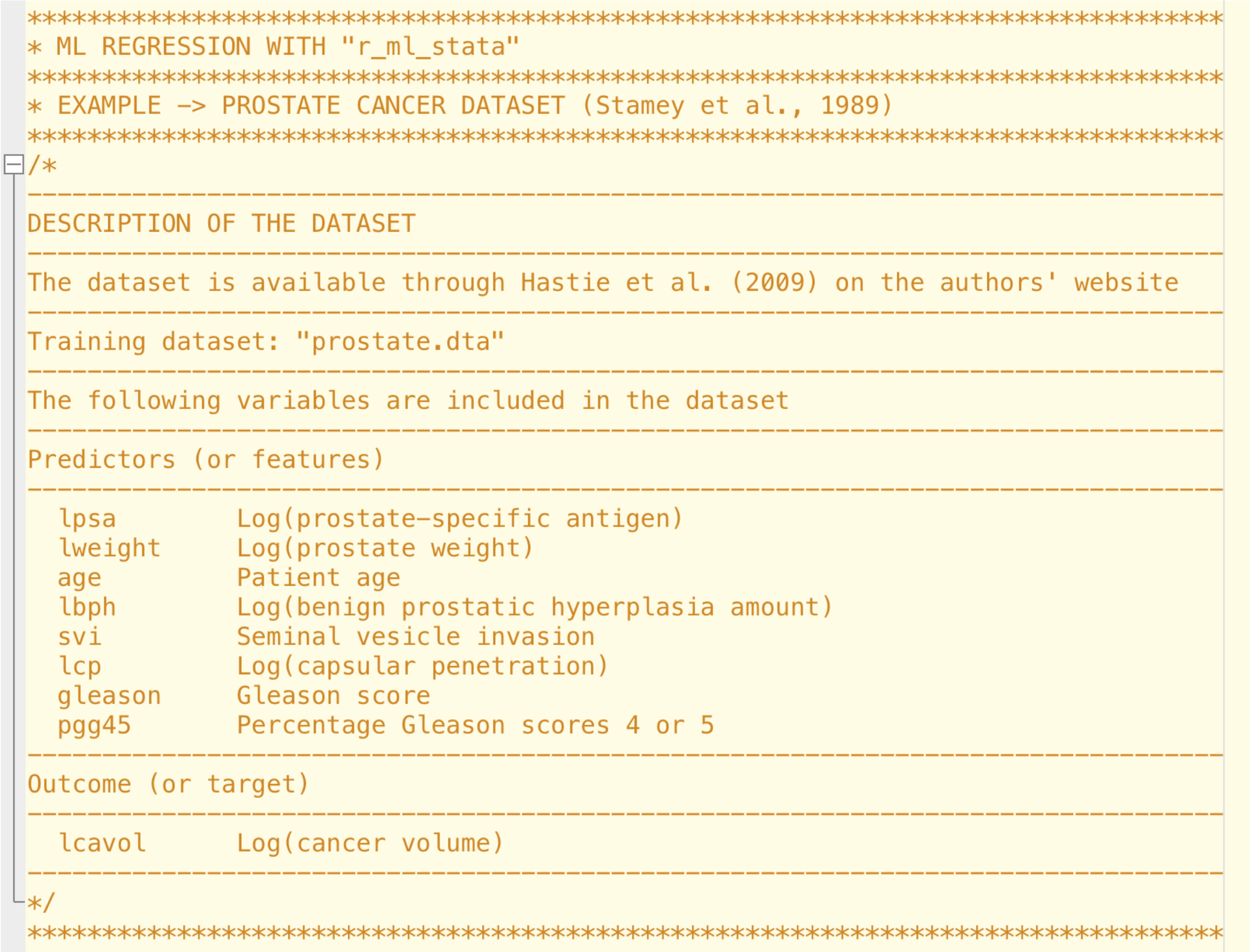


modeltype_options	Description
Model	
tree	Classification tree
randomforest	Bagging and random forests
boost	Boosting
regularizedmultinomial	Regularized multinomial
nearestneighbor	Nearest Neighbor
neuralnet	Neural network
naivebayes	Naive Bayes
svm	Support vector machine
multinomial	Standard multinomial

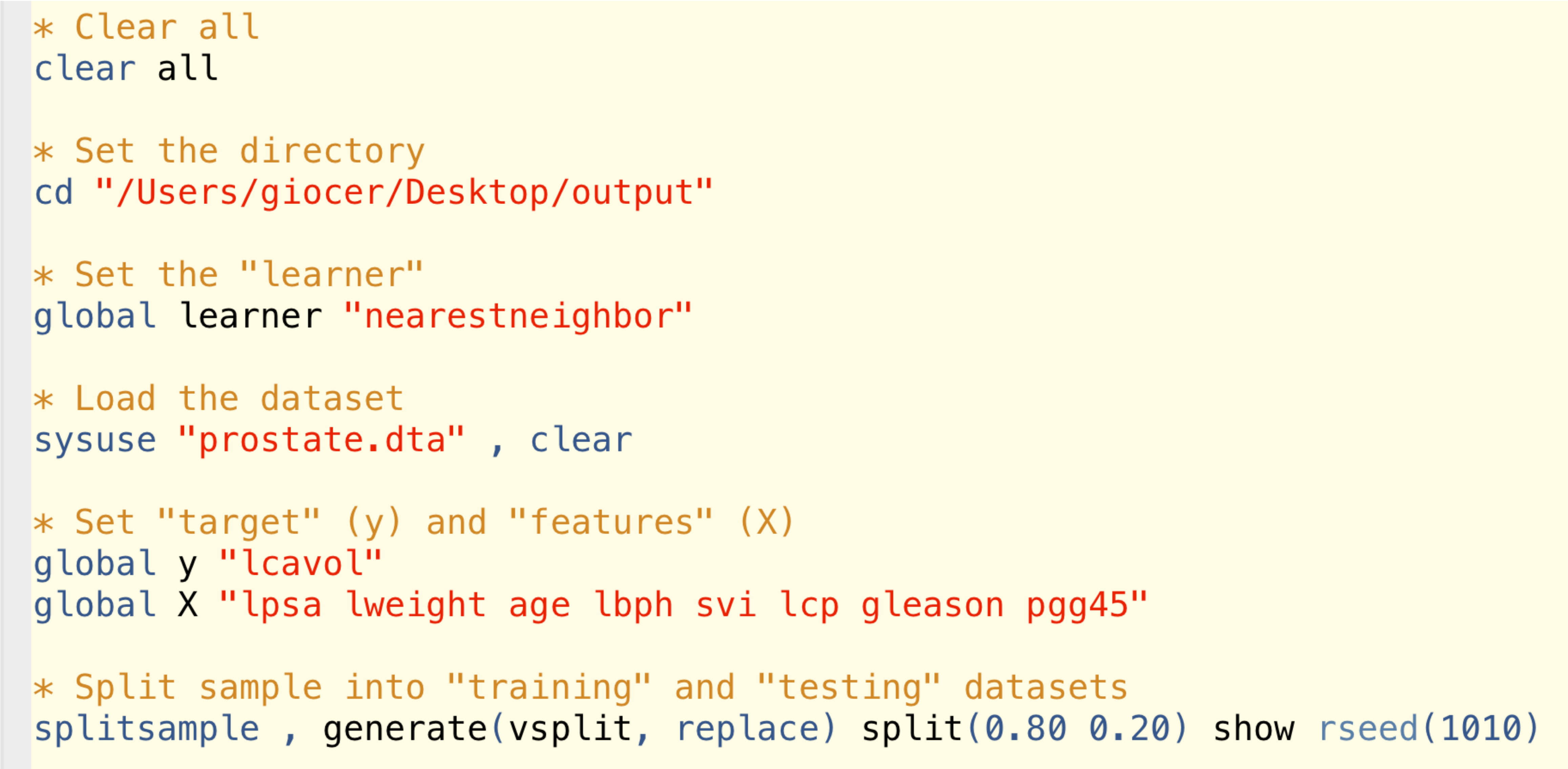
Classification

Practical implementation

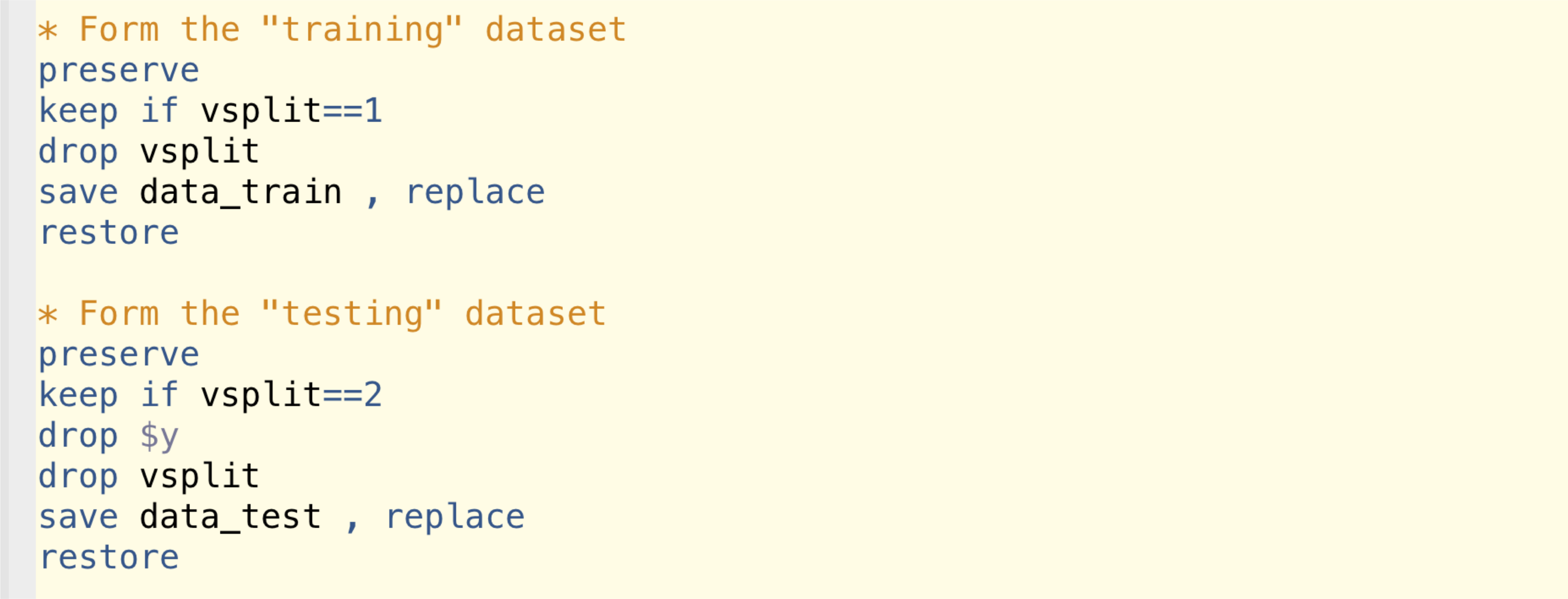
Nearest neighbor regression



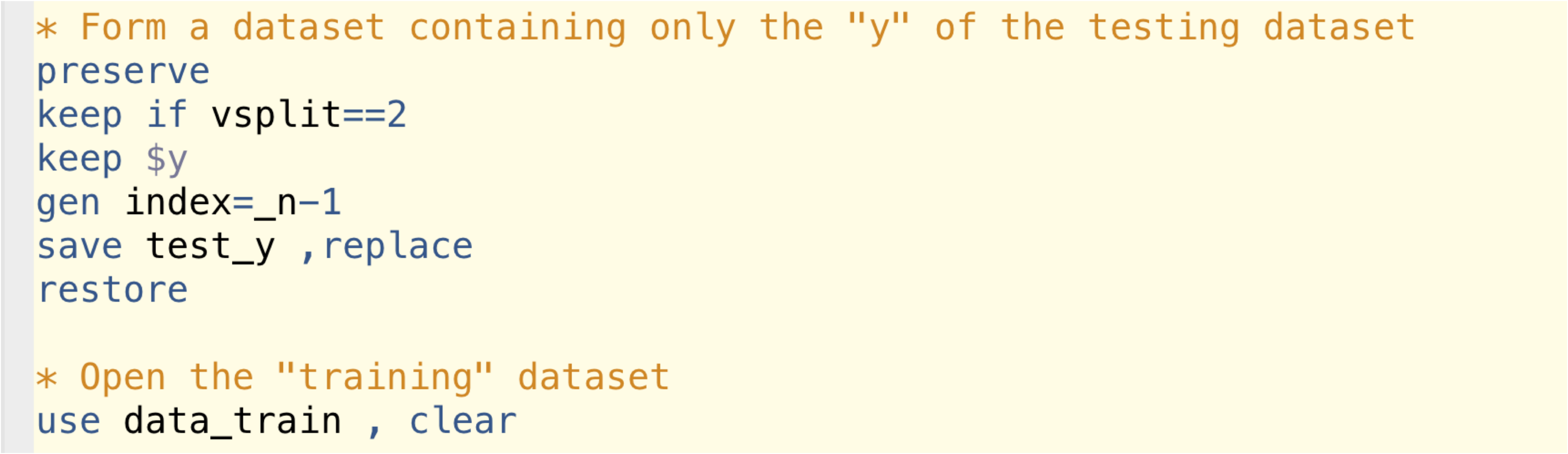
```
********************************************************************************
* ML REGRESSION WITH "r_ml_stata"
********************************************************************************
* EXAMPLE -> PROSTATE CANCER DATASET (Stamey et al., 1989)
********************************************************************************
/*
--------------------------------------------------------------------------------
DESCRIPTION OF THE DATASET
--------------------------------------------------------------------------------
The dataset is available through Hastie et al. (2009) on the authors' website
--------------------------------------------------------------------------------
Training dataset: "prostate.dta"
--------------------------------------------------------------------------------
The following variables are included in the dataset
--------------------------------------------------------------------------------
Predictors (or features)
--------------------------------------------------------------------------------
  lpsa        Log(prostate-specific antigen)
  lweight     Log(prostate weight)
  age         Patient age
  lbph        Log(benign prostatic hyperplasia amount)
  svi         Seminal vesicle invasion
  lcp         Log(capsular penetration)
  gleason     Gleason score
  pgg45       Percentage Gleason scores 4 or 5
--------------------------------------------------------------------------------
Outcome (or target)
--------------------------------------------------------------------------------
  lcavol      Log(cancer volume)
--------------------------------------------------------------------------------
*/
********************************************************************************
```



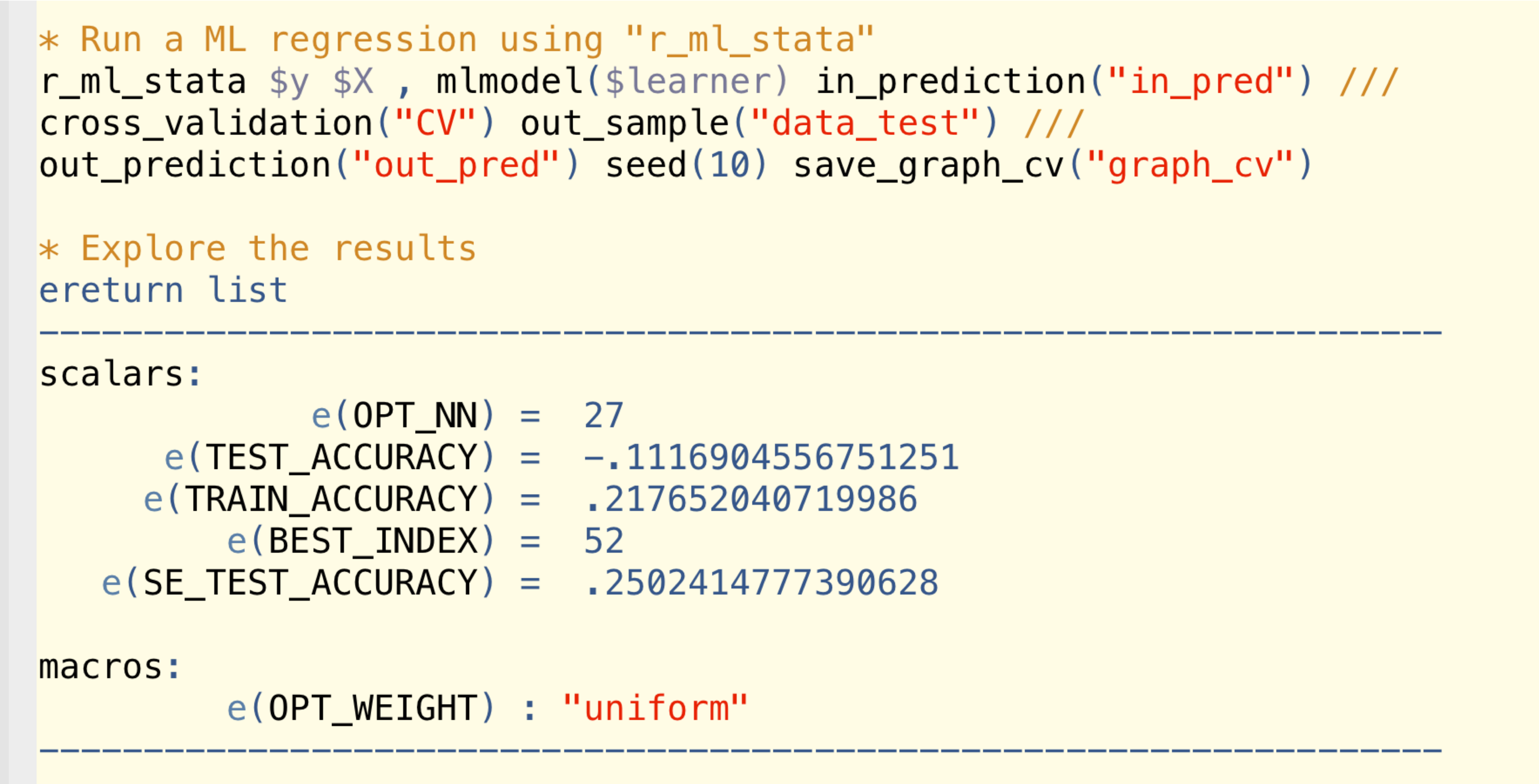
```
* Clear all
clear all

* Set the directory
cd "/Users/giocer/Desktop/output"

* Set the "learner"
global learner "nearestneighbor"

* Load the dataset
sysuse "prostate.dta" , clear

* Set "target" (y) and "features" (X)
global y "lcavol"
global X "lpsa lweight age lbph svi lcp gleason pgg45"

* Split sample into "training" and "testing" datasets
splitsample , generate(vsplit, replace) split(0.80 0.20) show rseed(1010)
```



```
* Form the "training" dataset
preserve
keep if vsplit==1
drop vsplit
save data_train , replace
restore

* Form the "testing" dataset
preserve
keep if vsplit==2
drop $y
drop vsplit
save data_test , replace
restore
```

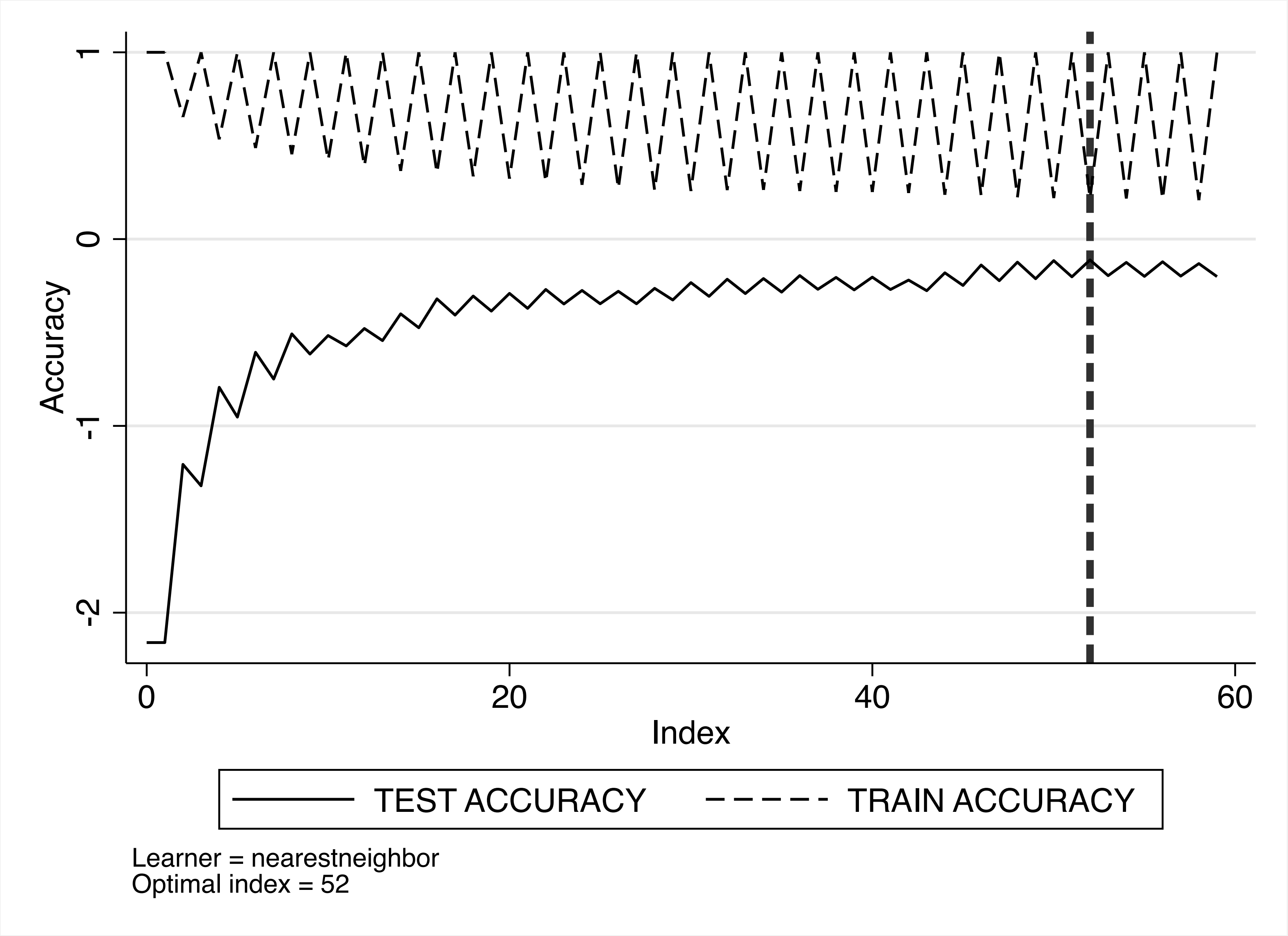


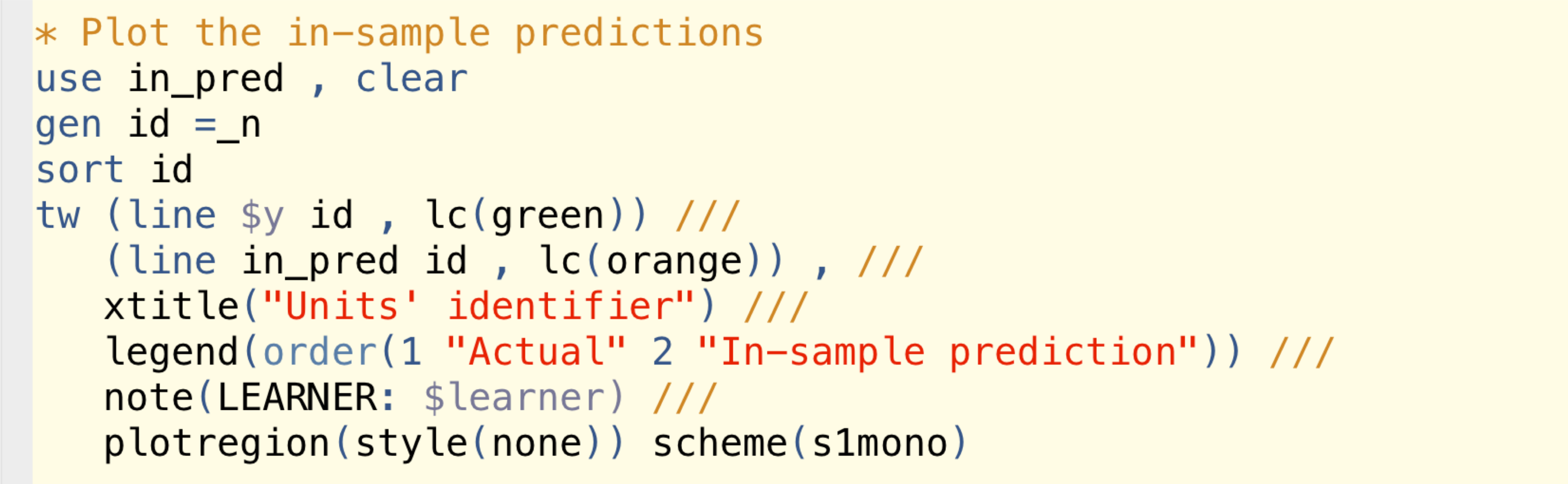
```
* Form a dataset containing only the "y" of the testing dataset
preserve
keep if vsplit==2
keep $y
gen index=_n-1
save test_y ,replace
restore

* Open the "training" dataset
use data_train , clear
```



```
* Run a ML regression using "r_ml_stata"
r_ml_stata $y $X , mlmodel($learner) in_prediction("in_pred") ///
cross_validation("CV") out_sample("data_test") ///
out_prediction("out_pred") seed(10) save_graph_cv("graph_cv")

* Explore the results
ereturn list
--------------------------------------------------------------------

scalars:
             e(OPT_NN) =  27
      e(TEST_ACCURACY) =  -.1116904556751251
     e(TRAIN_ACCURACY) =  .217652040719986
         e(BEST_INDEX) =  52
   e(SE_TEST_ACCURACY) =  .250241477390628

macros:
         e(OPT_WEIGHT) : "uniform"
--------------------------------------------------------------------
```

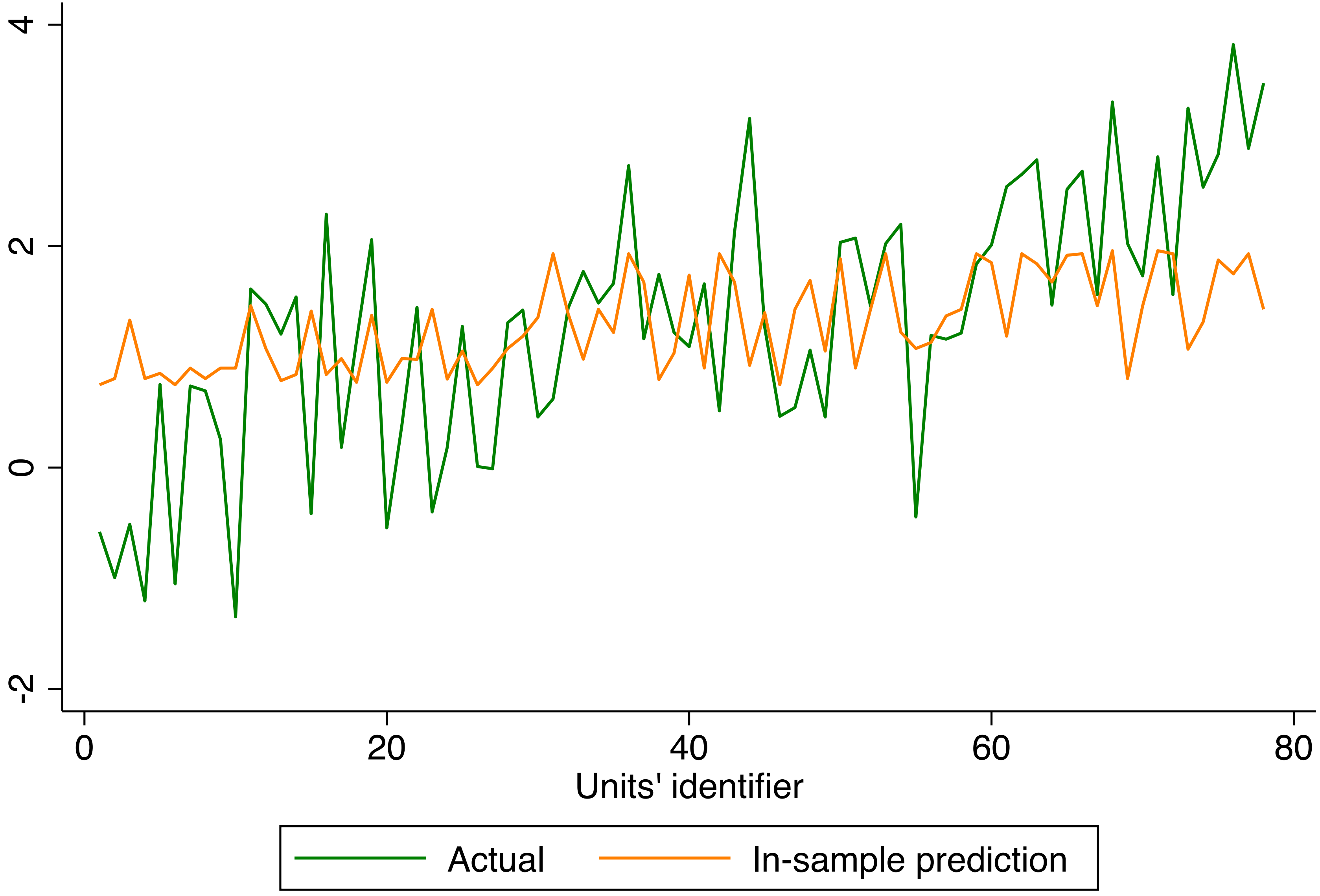
10-fold cross-validation results



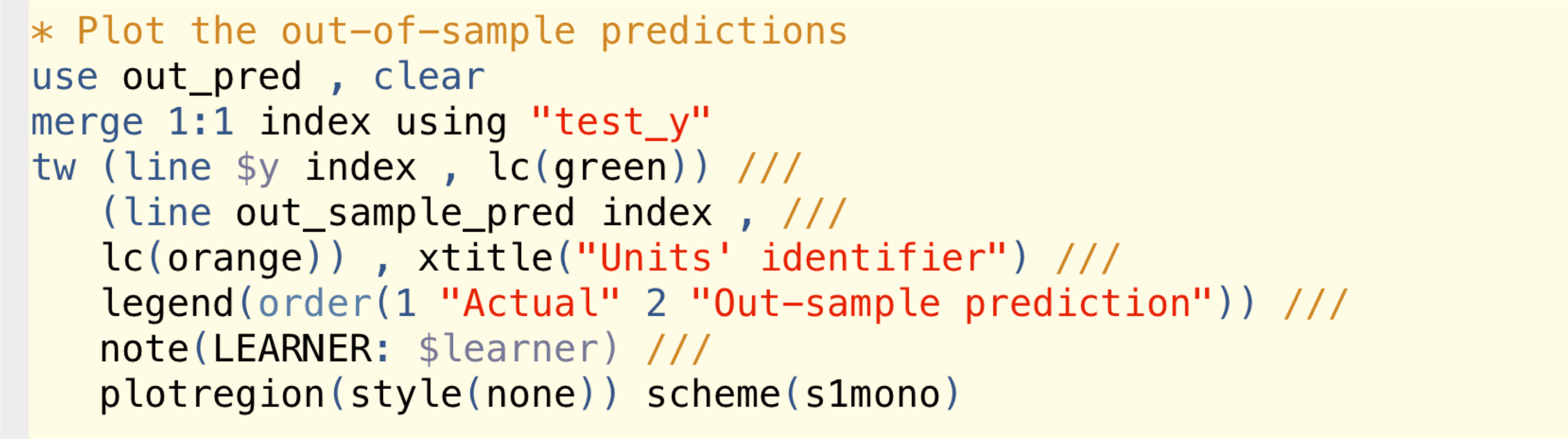


```
* Plot the in-sample predictions
use in_pred , clear
gen id =_n
sort id
tw (line $y id , lc(green)) ///
   (line in_pred id , lc(orange)) , ///
   xtitle("Units' identifier") ///
   legend(order(1 "Actual" 2 "In-sample prediction")) ///
   note(LEARNER: $learner) ///
   plotregion(style(none)) scheme(s1mono)
```

In-sample predictions

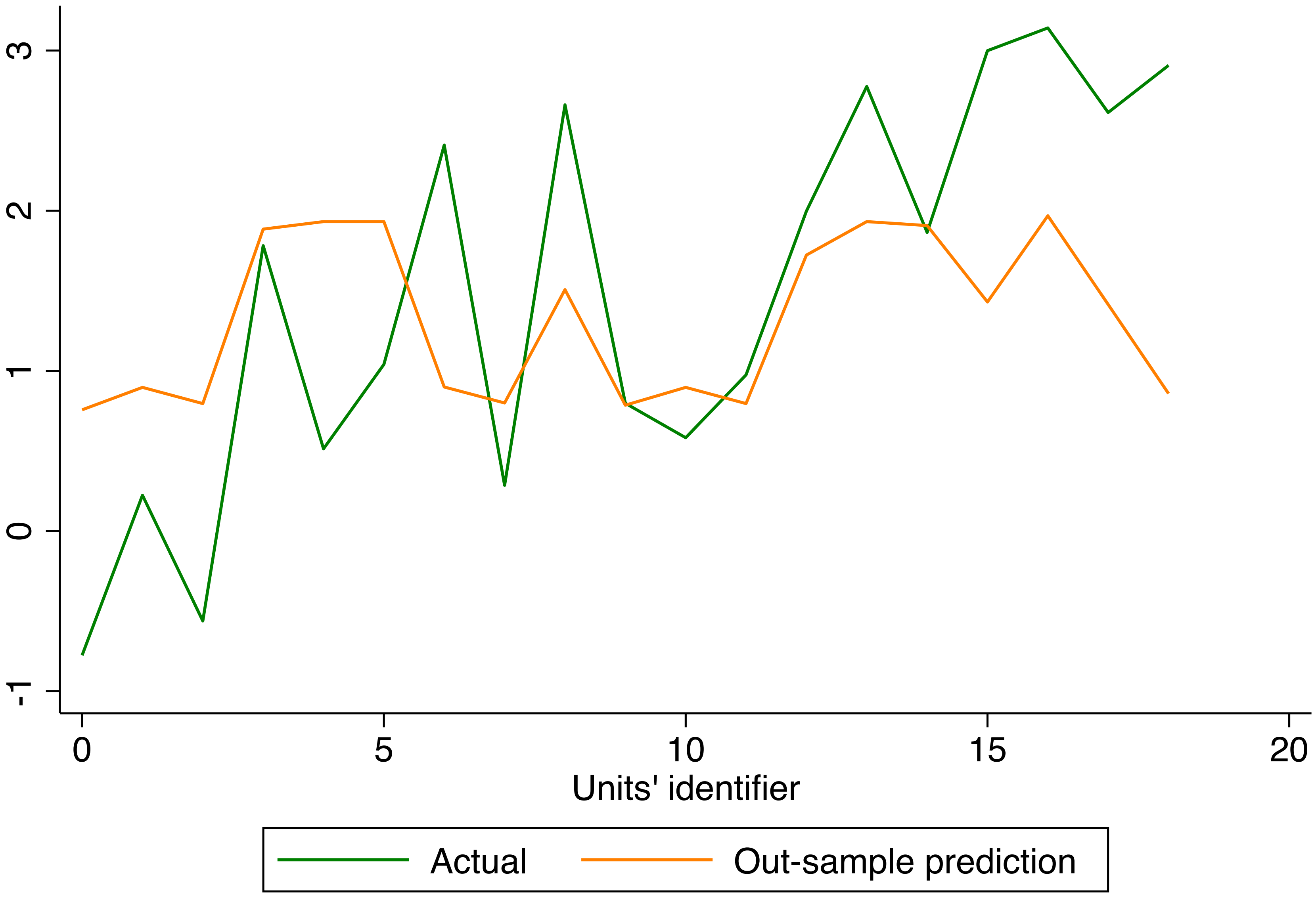


LEARNER: nearestneighbor



```
* Plot the out-of-sample predictions
use out_pred , clear
merge 1:1 index using "test_y"
tw (line $y index , lc(green)) ///
   (line out_sample_pred index , ///
   lc(orange)) , xtitle("Units' identifier") ///
   legend(order(1 "Actual" 2 "Out-sample prediction")) ///
   note(LEARNER: $learner) ///
   plotregion(style(none)) scheme(s1mono)
```

Out-of-sample prediction

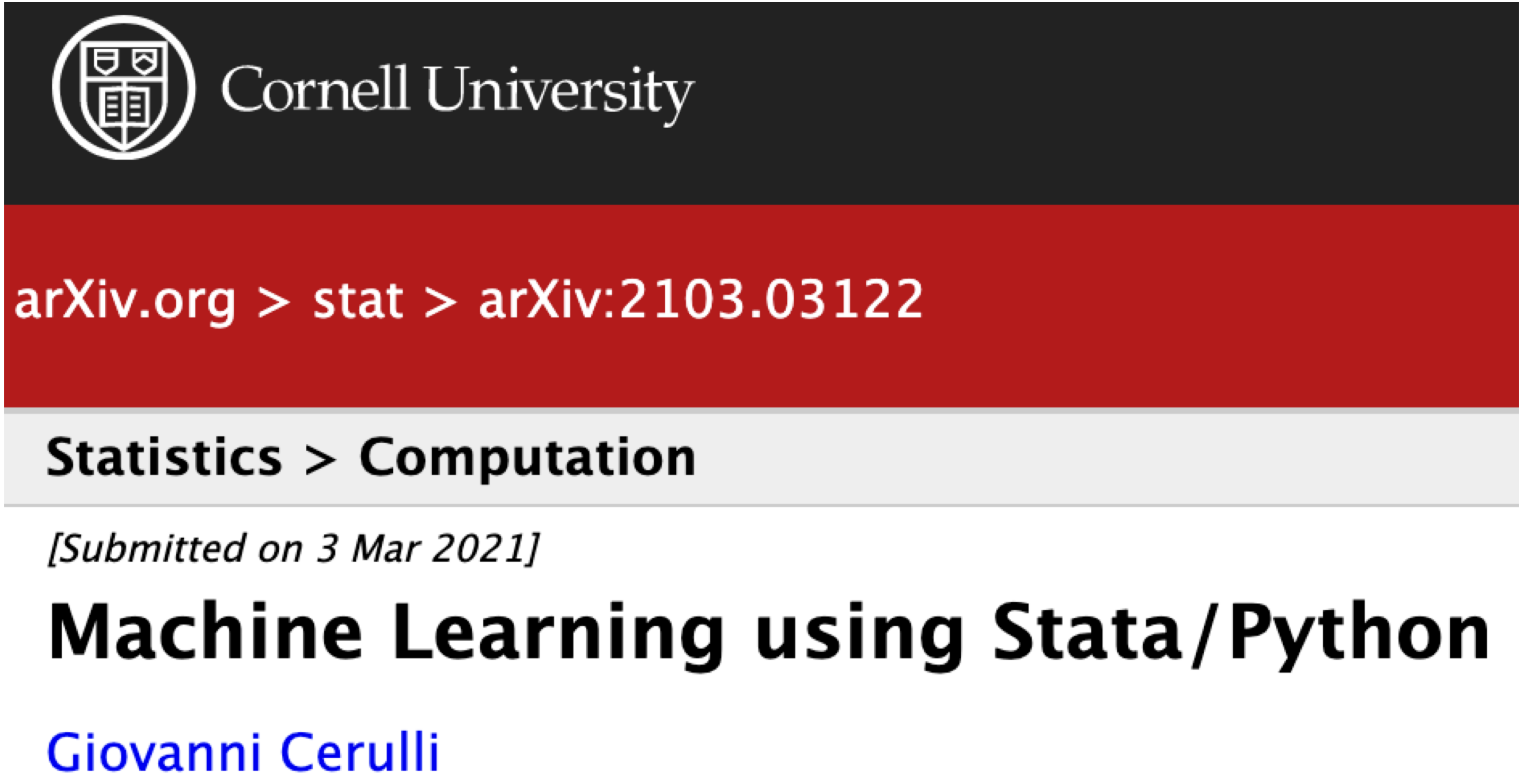


LEARNER: nearestneighbor

Example

Comparing multiple learners

Guessing whether a “new” car is a “foreign” or “domestic” one based on a series of characteristics, including price, number of repairs, weight, etc



Cornell University

arXiv.org > stat > arXiv:2103.03122

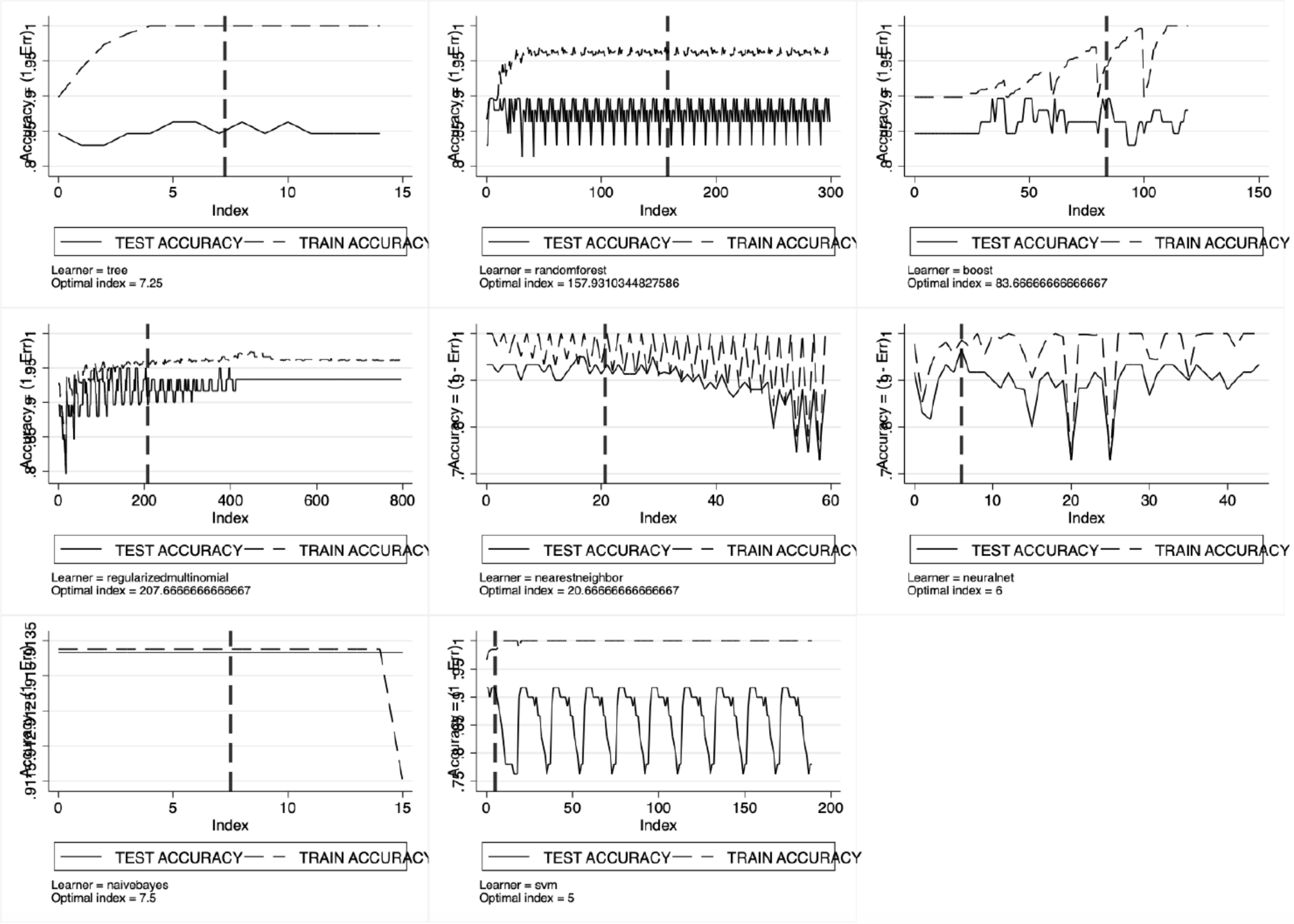
Statistics > Computation

[Submitted on 3 Mar 2021]

Machine Learning using Stata/Python

Giovanni Cerulli

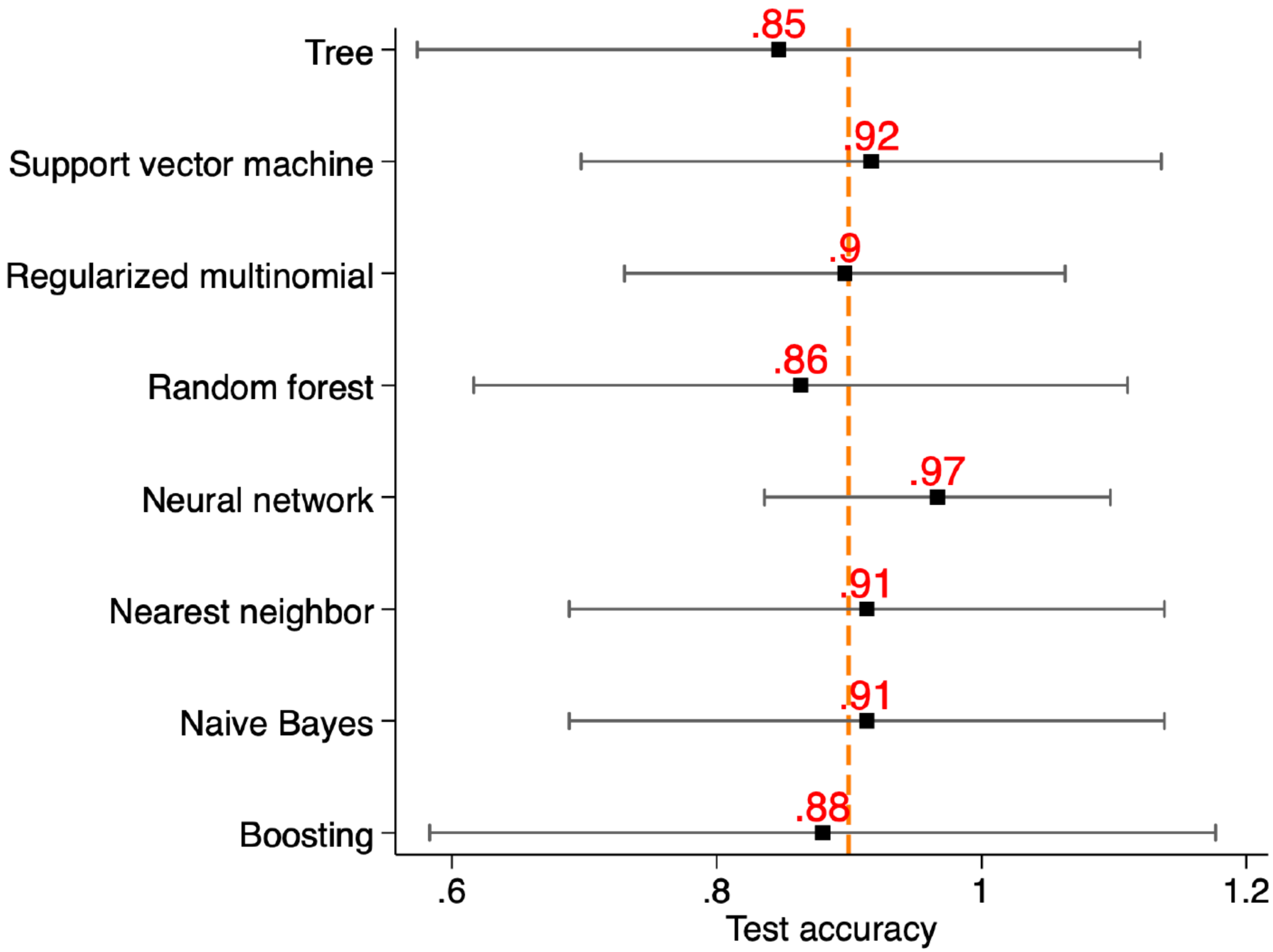
Cross-validation



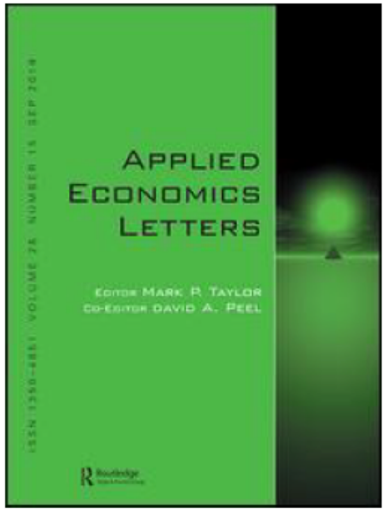
Cross-validation maximum of the classification test accuracy over a grid of learners' tuning parameters.

Accuracy measure: “error rate”

Comparing learner performance



Forest plot for comparing mean and standard deviation of different learners. Classification setting



APPLIED ECONOMICS LETTERS

Editor Mark P. Taylor
Co-Editor David A. Peel

Applied Economics Letters

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/rael20

Improving econometric prediction by machine learning

Giovanni Cerulli

References

- Cerulli, G. 2020. *C_ML_STATA: Stata module to implement machine learning classification in Stata*. Statistical Software Components, Boston College Department of Economics. Available at: https://econpapers.repec.org/software/bocbocode/s458830.htm
- Cerulli, G. 2020. *R_ML_STATA: Stata module to implement machine learning regression in Stata*. Statistical Software Components, Boston College Department of Economics. Available at: https://econpapers.repec.org/software/bocbocode/s458831.htm
- Cerulli, G. 2020. *A super-learning machine for predicting economic outcomes*, MPRA Paper 99111, University Library of Munich, Germany, 2020
- Cerulli, G. 2020. Improving econometric prediction by machine learning, *Applied Economics Letters*, Forthcoming.
- Gareth, J., Witten, D., Hastie, D.T., Tibshirani, R. 2013. *An Introduction to Statistical Learning : with Application in R*. New York, Springer
- Raschka, S., Mirjalili, V. 2019. *Python Machine Learning*. 3rd Edition, Packt Publishing.