

# MLJ : Notes

---

Dainish Jabeen

January 12, 2023

## 1 General

MLJ is a machine learning toolbox for Julia, that wraps a large number of models and provides great tools such as resampling and evaluation [1] [2].

Models are structs storing hyperparameters.

Listing 1: Basics

```
1
2      #Split data randomly with seed (rng)
3      y, X = unpack(df,==( :col ); rng=123);
4
5      df |> pretty #Output pretty version
6
7      #Load model
8      VAR = @load model pkg=""
9
10     var = VAR() #Default parameters
11
12     #evaluate model with cross validation
13     via error measures
14
15     evaluate(var,X,y,resampling=CV(shuffle=true),measures=[rms])
```

### 1.1 Types

Each model has an expected variable type needed to train it:

**target\_scitype(model)** provides type needed.

Listing 2: Change type

```
1
2      #Change type of var
3
4      y = coerce(df, :col=type,...)
```

## 2 Machines

Used to store training outcomes.

Listing 3: Machines

```
1      mach = machine(model,X,y)
2
3
4      #70:30 partition giving an index vector
5      train , test = partition(eachindex(y),0.7)
6
7      fit!(mach,train)
8      yhat = predict(mach,X[test,:]);
9
10     #error rate
11     misclassification_rate(yhat, test)
```

Can also evaluate! machines.

### 2.1 prediction

To predict with a given probability for a class use **broadcast**.

For a matrix of all classes: **pdf**.

For an output of a class: **predict\_mode**.

## 3 Hyperparameter tuning

- Create ensemble model
- Create ranges for parameters
- wrap in tuned model
- create mach, fit data
- report on best model
- mach will become the best model

**learning\_curve()** gives a performance line for a tuning parameter.

## 4 Pipeline

Linear set of models chained together, that can be evaluated and used as a single model.

For example: Change var type -> tune parameter -> model -> train.

## References

- [1] A. D. Blaom, F. Kiraly, T. Lienart, Y. Simillides, D. Arenas, and S. J. Vollmer, “MLJ: A julia package for composable machine learning,” *Journal of Open Source Software*, vol. 5, no. 55, p. 2704, 2020. DOI: 10.21105/joss.02704. [Online]. Available: <https://doi.org/10.21105/joss.02704>.
- [2] A. D. Blaom and S. J. Vollmer, *Flexible model composition in machine learning and its implementation in MLJ*, 2020. arXiv: 2012.15505 [cs.LG].