

01_H2o_AutoML

September 28, 2018

1 This tutorial will teach you how to create a ML model using H2o

```
In [4]: install.packages("h2o")
```

Installing package into /home/nbuser/R
(as lib is unspecified)

```
In [5]: library(h2o)
```

Your next step is to start H2O:

```
> h2o.init()
```

For H2O package documentation, ask for help:

```
> ??h2o
```

After starting H2O, you can use the Web UI at <http://localhost:54321>

For more information visit <http://docs.h2o.ai>

Attaching package: h2o

The following objects are masked from package:stats:

```
cor, sd, var
```

The following objects are masked from package:base:

```
&&, %*%, %in%, ||, apply, as.factor, as.numeric, colnames,  
colnames<-, ifelse, is.character, is.factor, is.numeric, log,  
log10, log1p, log2, round, signif, trunc
```

```
In [4]:
```

```
Installing package into /home/nbuser/R  
(as lib is unspecified)
```

```
In [6]: h2o.init()
```

```
H2O is not running yet, starting it now...
```

```
Note: In case of errors look at the following log files:
```

```
/tmp/RtmpJPHRuK/h2o_nbuser_started_from_r.out  
/tmp/RtmpJPHRuK/h2o_nbuser_started_from_r.err
```

```
Starting H2O JVM and connecting: . Connection successful!
```

```
R is connected to the H2O cluster:
```

```
H2O cluster uptime:      4 seconds 564 milliseconds  
H2O cluster timezone:    Etc/UTC  
H2O data parsing timezone: UTC  
H2O cluster version:     3.20.0.8  
H2O cluster version age: 6 days  
H2O cluster name:        H2O_started_from_R_nbuser_ebt489  
H2O cluster total nodes: 1  
H2O cluster total memory: 0.85 GB  
H2O cluster total cores: 2  
H2O cluster allowed cores: 2  
H2O cluster healthy:     TRUE  
H2O Connection ip:        localhost  
H2O Connection port:      54321  
H2O Connection proxy:     NA  
H2O Internal Security:    FALSE  
H2O API Extensions:       XGBoost, Algos, AutoML, Core V3, Core V4  
R Version:                R version 3.4.1 (2017-06-30)
```

```
In [7]: # Import a sample binary outcome train/test set into H2O
```

```
train <- h2o.importFile("https://raw.githubusercontent.com/caiomsouza/ml-open-datasets/master/data/train/train.csv")
```

```
|=====| 100%
```

```
In [8]: test <- h2o.importFile("https://raw.githubusercontent.com/caiomsouza/ml-open-datasets/master/data/test/test.csv")
```

```
|=====| 100%
```

```
In [11]: aml <- h2o.automl(y = "TARGET", training_frame = train, max_runtime_secs = 60)
```

```
lb <- aml@leaderboard
```

```
|=====| 100%
```

```
In [12]: pred <- h2o.predict(aml@leader, test)
```

```
pred.df <- as.data.frame(pred)
```

```
|=====| 100%
```

```
In [13]: pred.df
```

predict
0.03018201
0.06007185
0.01234840
0.08411874
0.01219768
0.10518281
0.07932790
0.11561808
0.02750700
0.03638500
0.02499721
0.01210655
0.02348450
0.01353224
0.01226373
0.08216215
0.04625748
0.01220058
0.01312592
0.04499044
0.03076510
0.10969968
0.01246462
0.01208948
0.16685433
0.01870491
0.01235863
0.01258144
0.05712224
0.01207897
...
0.01204541
0.02203033
0.01212965
0.02524231
0.01316983
0.02402205
0.09429323
0.17664539
0.01308555
0.05003763
0.04497611
0.02153374
0.02992880
0.02227056
0.24198404
0.13621792
0.01424813
0.02770244
0.11230710
0.01276678
0.08890251

```

In [14]: write.csv(pred.df, file = "pred_h2o_automl.csv")

In [15]: testIds<-as.data.frame(test$ID)
         submission<-data.frame(cbind(testIds,pred.df$predict))
         colnames(submission)<-c("ID", "PredictedProb")

In [16]: write.csv(submission, "pred_h2o_automl_with_ID.csv", row.names=T)

In [17]: write.csv(submission, "pred_h2o_automl_with_ID_no_Row_name.csv", row.names=F)

In [18]: lb

```

				model_id	mean_residual_deviance
1				StackedEnsemble_AllModels_0_AutoML_20180928_110851	0.03525777
2				StackedEnsemble_BestOfFamily_0_AutoML_20180928_110851	0.03534037
3				DRF_0_AutoML_20180928_104413	0.03633521
4				DRF_0_AutoML_20180928_110851	0.03643848
	rmse	mse	mae	rmsle	
1	0.1877705	0.03525777	0.07050069	0.1308395	
2	0.1879904	0.03534037	0.07066108	0.1309889	
3	0.1906180	0.03633521	0.07213644	0.1354914	
4	0.1908886	0.03643848	0.07223831	0.1357260	

[4 rows x 6 columns]

```

In [23]: lb$rmse

```

	rmse
1	0.1877705
2	0.1879904
3	0.1906180
4	0.1908886

[4 rows x 1 column]

```

In [24]: lb$mse

```

	mse
1	0.03525777
2	0.03534037
3	0.03633521
4	0.03643848

[4 rows x 1 column]