Deep Learning & AutoML in h2o

Set up

```
# Load libraries
library(h2o)
library(tidyverse)
library(wesanderson)
library(knitr)

# Disable progress bar in document
h2o.no_progress()

# Start h2o cluster
h2o.init(nthreads = -1,
         max_mem_size = '4G')
```

```
##  Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         3 hours 1 seconds
##     H2O cluster timezone:       Europe/Warsaw
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.40.0.1
##     H2O cluster version age:    1 month and 6 days
##     H2O cluster name:           H2O_started_from_R_User_acc441
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   3.68 GB
##     H2O cluster total cores:    16
##     H2O cluster allowed cores:  16
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     R Version:                  R version 4.2.2 (2022-10-31 ucrt)
```

```r
# Load file
mushrooms <- read.csv('https://tinyurl.com/hmkhs9au')

# Transform data set for further analysis
mushrooms <- mushrooms %>%
  # Remove not needed characters
  mutate(across(1:23, ~ substr(.x, 3,3))) %>%
  # Change columns from strings to factors
  mutate(across(everything(), as.factor))

# Load data to h2o cluster
mushrooms_hex <- as.h2o(mushrooms, destination_frame = 'mushrooms_hex')

# Split data set to train (75%) & test (25%)
mushrooms_split <- h2o.splitFrame(data = mushrooms_hex, ratios = 0.75)

mushrooms_train <- mushrooms_split[[1]]
mushrooms_test <- mushrooms_split[[2]]
```

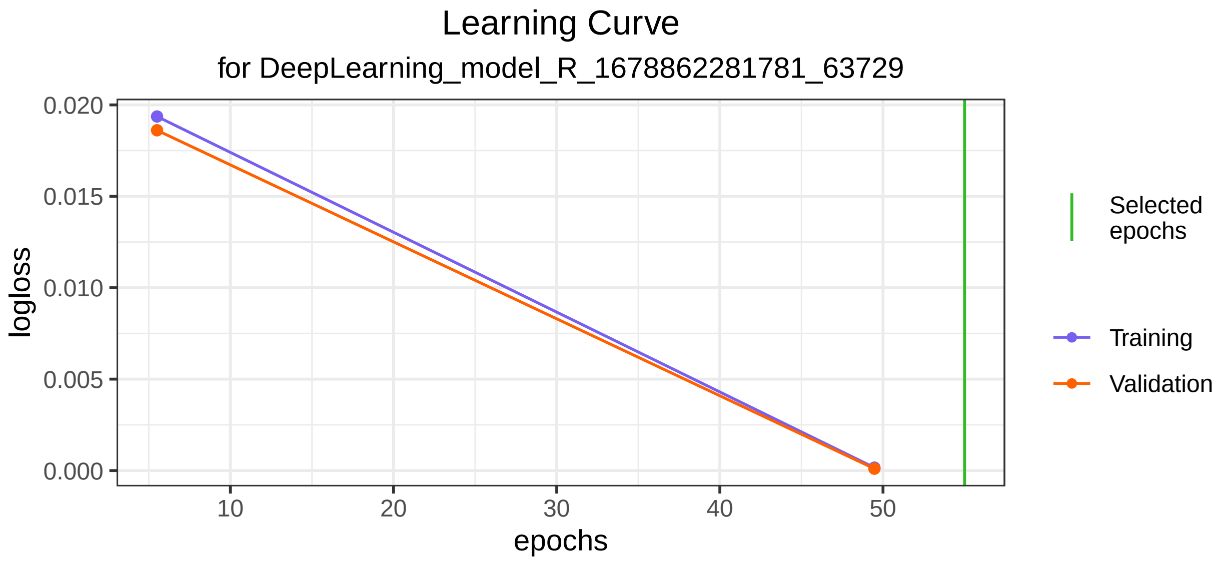
Data

Data set contains 8124 observations of 23 species from Agaricus and Lepiota families.
Beside class (e - edible/p - poisonous or uknown edibility) there are 22 physical attributes.

h2o's Deep Learning

```r
# Build and train Deep Learning model
mushrooms_dl <- h2o.deeplearning(
                        y = 23,
                        x = 1:22,
                        training_frame = mushrooms_train,
                        validation_frame = mushrooms_test,
                        distribution = 'multinomial',
                        activation = 'RectifierWithDropout',
                        hidden = c(100, 200, 100),
                        input_dropout_ratio = 0.2,
                        l1 = 1e-5,
                        epochs = 55,
                        variable_importances = TRUE)
```

```r
# Learning curve plot
h2o.learning_curve_plot(mushrooms_dl)
```



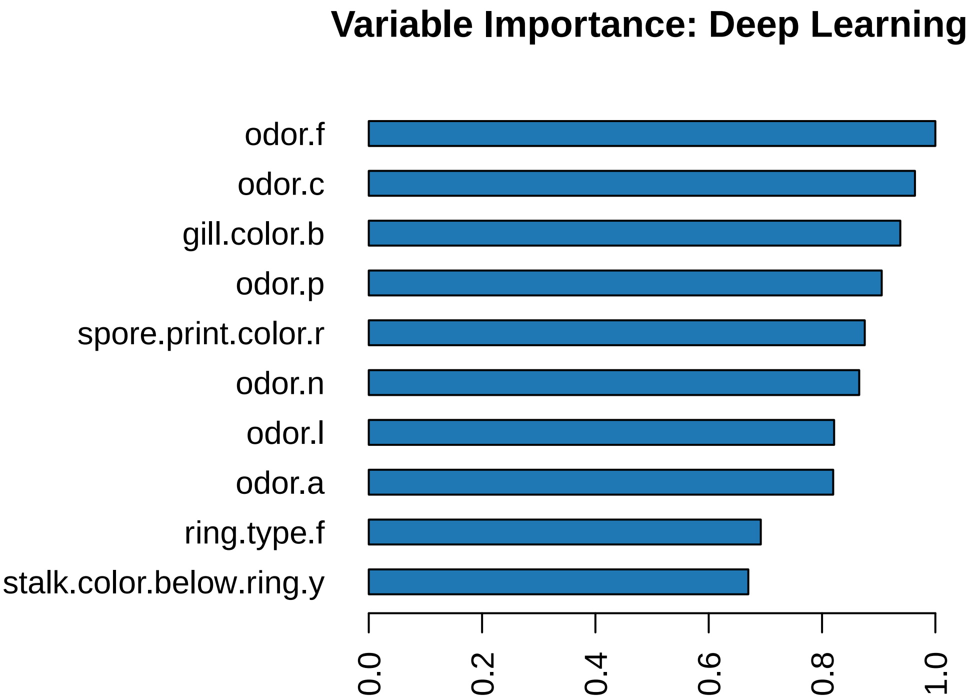
```r
# Confusion matrix
h2o.confusionMatrix(mushrooms_dl, mushrooms_test, valid = FALSE, xval = FALSE)
```

```
## Confusion Matrix (vertical: actual; across: predicted)  for max f1 @ threshold = 0.86974572554556:
##           e    p    Error     Rate
## e      1016    0 0.000000  =0/1016
## p         0 1004 0.000000  =0/1004
## Totals 1016 1004 0.000000  =0/2020
```

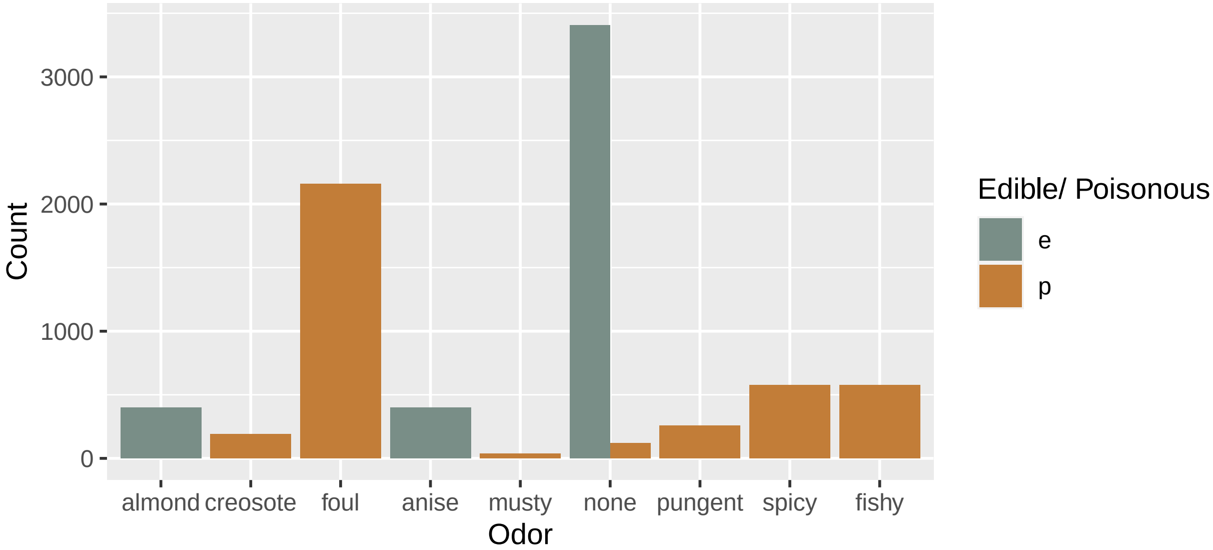
```r
# If needed, predictions can be done on test data set
# (which was used for validation in model training)
mushroom_pred <- h2o.predict(object = mushrooms_dl,
                             newdata = mushrooms_test)

# Confusion matrix as simple table: test vs predictions
table(as.data.frame(mushrooms_test[,23])[,1],
      as.data.frame(mushroom_pred[,1])[,1])
```

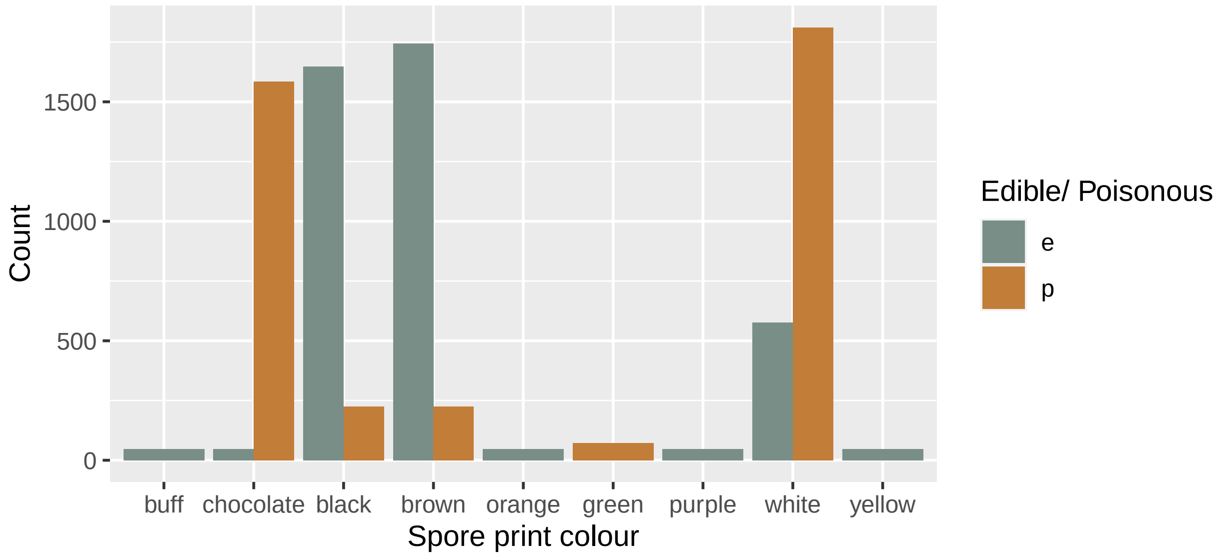
```
##
##         e    p
##    e 1016    0
##    p    0 1004
```

```
# Importance of parameters
h2o.varimp_plot(mushrooms_dl)
```



Most important parameters are odor and spore print color:





AutoML

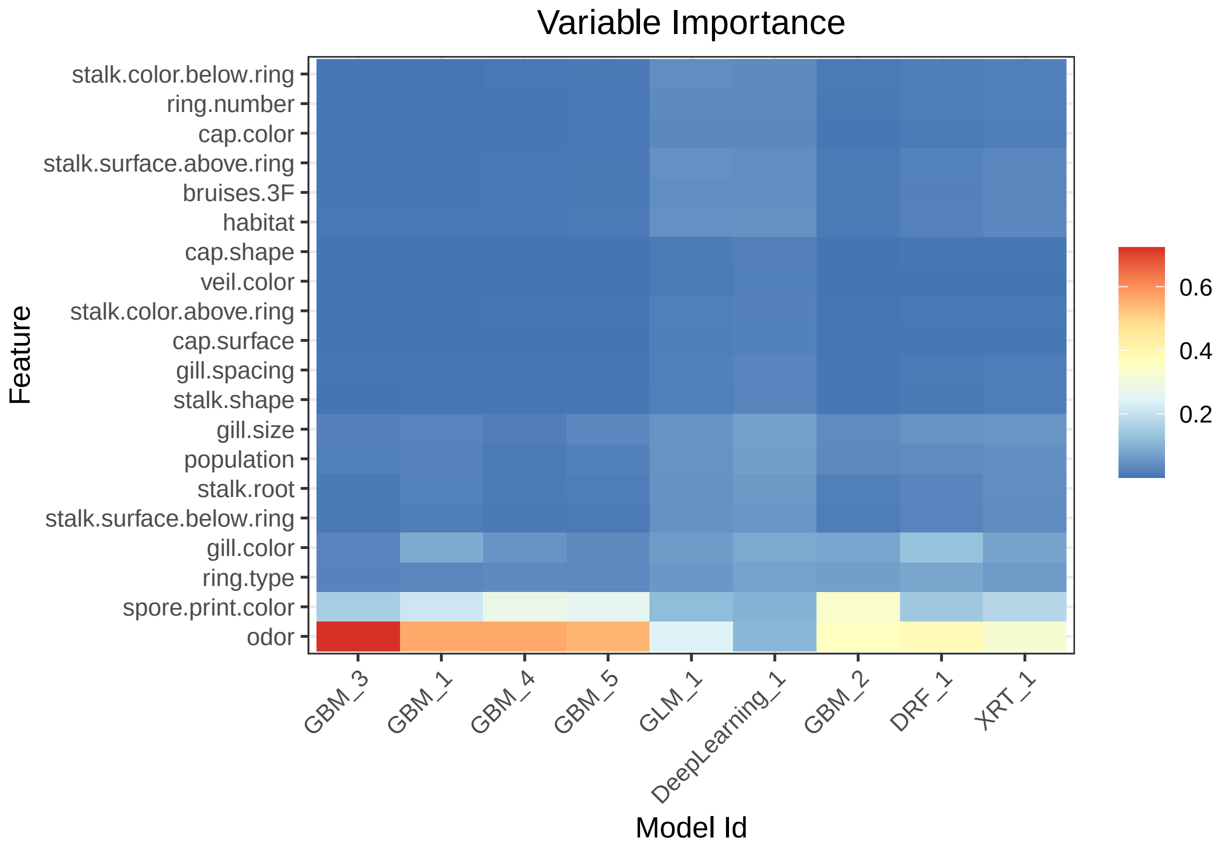
```
# Run AutoML: max 100 models, max 60 seconds
mushrooms_auoml <- h2o.automl(
                        y = 23,
                        x = 1:22,
                        training_frame = mushrooms_train,
                        max_runtime_secs = 60,
                        max_models = 100)
```

```
##
## 10:38:26.963: AutoML: XGBoost is not available; skipping it.
## 10:38:26.968: _train param, Dropping bad and constant columns: [veil.type]
## 10:38:27.481: _train param, Dropping bad and constant columns: [veil.type]
## 10:38:40.666: _train param, Dropping bad and constant columns: [veil.type]
## 10:38:41.55: _train param, Dropping bad and constant columns: [veil.type]
## 10:38:48.666: _train param, Dropping bad and constant columns: [veil.type]
## 10:38:56.890: _train param, Dropping bad and constant columns: [veil.type]
## 10:39:05.553: _train param, Dropping bad and constant columns: [veil.type]
## 10:39:06.108: _train param, Dropping bad and constant columns: [veil.type]
## 10:39:12.414: _train param, Dropping bad and constant columns: [veil.type]
```

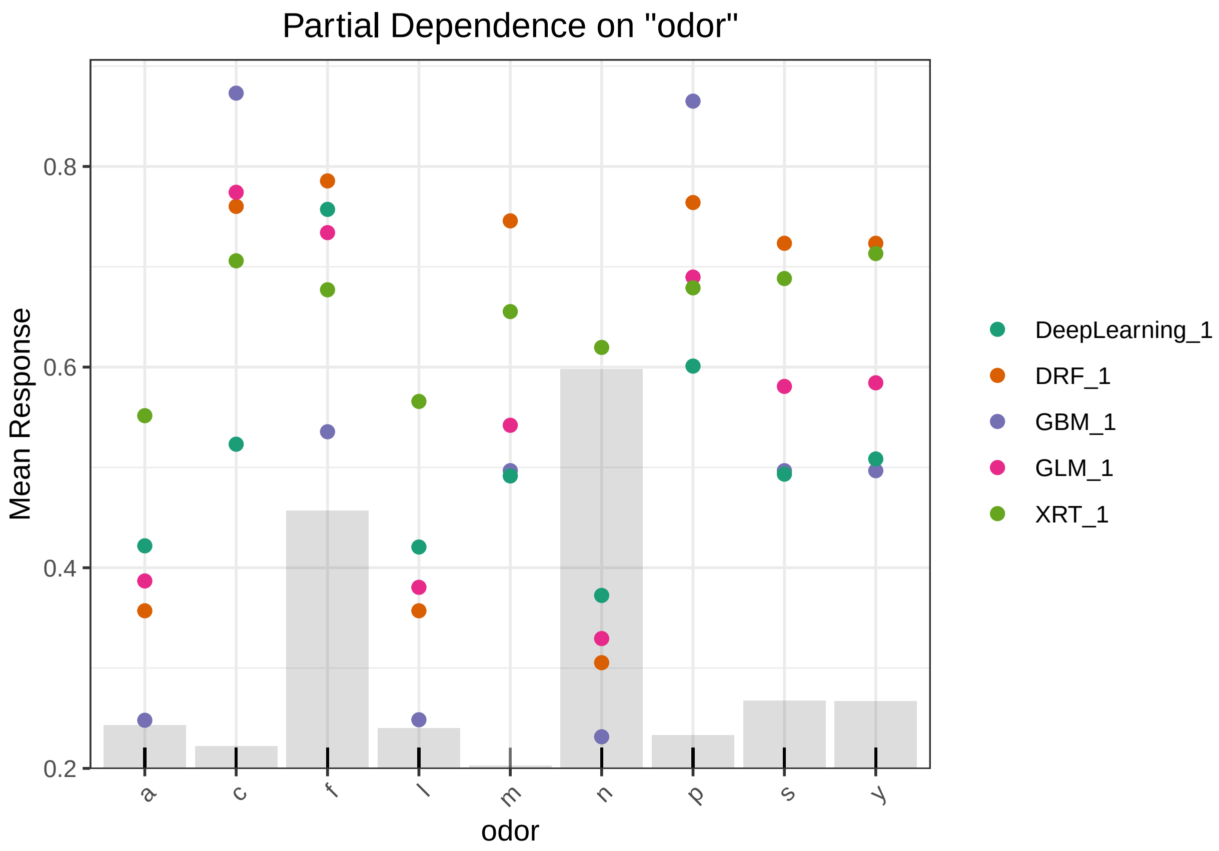
```
# Leader board
df <- h2o.get_leaderboard(object = mushrooms_auoml, extra_columns = "ALL")
df <- as.data.frame(df)
df$model_id <- substr(df$model_id,1,5)
df <- df %>% mutate(across(where(is.numeric), round, 3))
df <- df[,-10]
kable(df,
      col.names = c('Model', 'auc', 'Log Loss', 'aucPR', 'Mean per class err.', 'RMSE',
                    'MSE', 'Train. time [ms]', 'Predict time / row [ms]'))
```

Model	auc	Log Loss	aucPR	Mean per class err.	RMSE	MSE	Train. time [ms]	Predict time / row [ms]
DRF_1	1	0.005	1	0.000	0.013	0.000	91	0.004
GBM_1	1	0.000	1	0.000	0.000	0.000	3587	0.116
GBM_2	1	0.000	1	0.000	0.000	0.000	2069	0.065
GBM_4	1	0.000	1	0.000	0.000	0.000	2318	0.074
GBM_3	1	0.000	1	0.000	0.000	0.000	2144	0.069
GBM_5	1	0.000	1	0.000	0.008	0.000	1623	0.055
GLM_1	1	0.001	1	0.000	0.007	0.000	110	0.002
DeepL	1	0.001	1	0.000	0.018	0.000	224	0.005
XRT_1	1	0.313	1	0.001	0.301	0.091	115	0.005

```
# Variables importance heatmap for different AutoML models
h2o.varimp_heatmap(mushrooms_auoml)
```



```r
# Effect of odor variable for each model
h2o.pd_multi_plot(mushrooms_auoml, mushrooms_test, "odor")
```



References

1. **Data set**: https://www.kaggle.com/datasets/ulrikthygepedersen/mushroom-attributes
2. **Agaricus family graphic**: https://en.wikipedia.org/wiki/Agaricus
3. **Mushrooms graphics**: <https://biolwww.usask.ca/fungi/glossary.html>
4. **h2o Deep Learning**: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html
5. **h2o AutoML**: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html
6. **h2o models explainability**: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/explain.html#explanation-plotting-functions