Regression: Predicting Quality of Red Wine

Prithviraj Lakkakula

1/23/2022

Contents

Data Source and Information

Data are collected from https://archive.ics.uci.edu/ml/datasets/wine+quality. Originally, there are two datasets, including red wine and white wine. However, in this project, I consider only the red wine data set. The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. See [Cortez et al., 2009] for more details. In the source, there was a note that due to privacy and logistical issues, only physicochemical (inputs) and sensory (the output) variables are available, which means that there may be other variables that could be missing that could have helped our prediction to be more accurate than I could predict in this analysis. Therefore, bear in mind that there could be omitted variable bias in this analysis. The source mentioned that other omitted variables could include grape type, wine brand, wine selling price etc.

The raw dataset for red wine contains a total of 12 variables, including quality (score between 0 and 1), which is our response varible in our regression setting and 11 other physicochemical attributes as inputs. These include fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.

Data Preprocessing

Data Summary

```
red_wine <- read.csv("winequality-red.csv")
summary(red_wine)
```

```
##  fixed_acidity   volatile_acidity  citric_acid    residual_sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free_sulfur_do2 tot_sulfur_do2      density
##  Min.   :0.01200   Min.   : 1.00   Min.   :  6.00   Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00   1st Qu.: 22.00   1st Qu.:0.9956
##  Median :0.07900   Median :14.00   Median : 38.00   Median :0.9968
##  Mean   :0.08747   Mean   :15.87   Mean   : 46.47   Mean   :0.9967
##  3rd Qu.:0.09000   3rd Qu.:21.00   3rd Qu.: 62.00   3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00   Max.   :289.00   Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

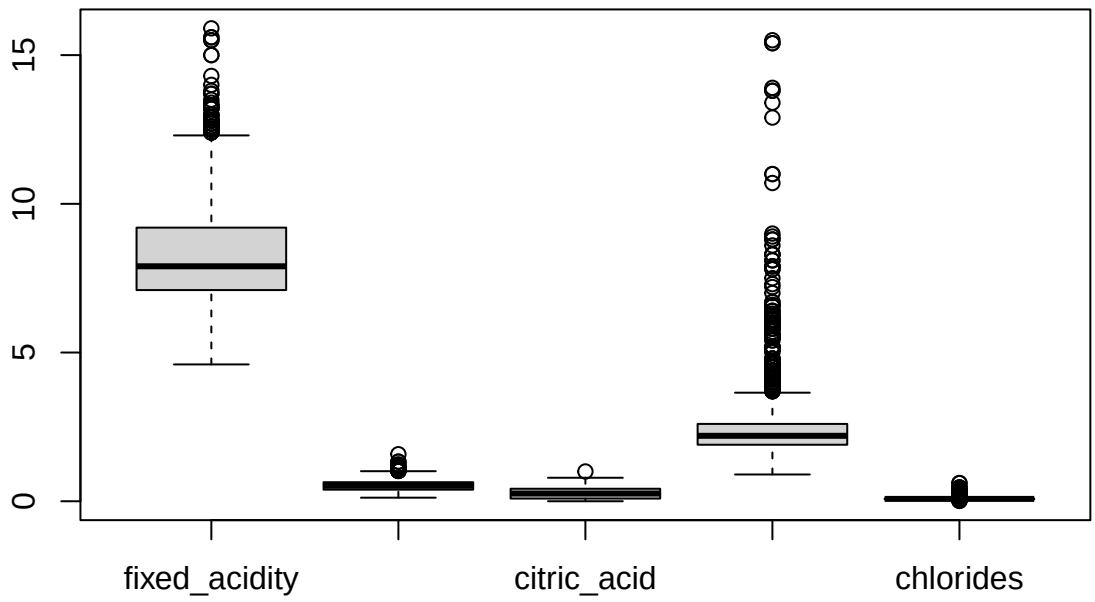
```
#str(red_wine)
```

Missing Values

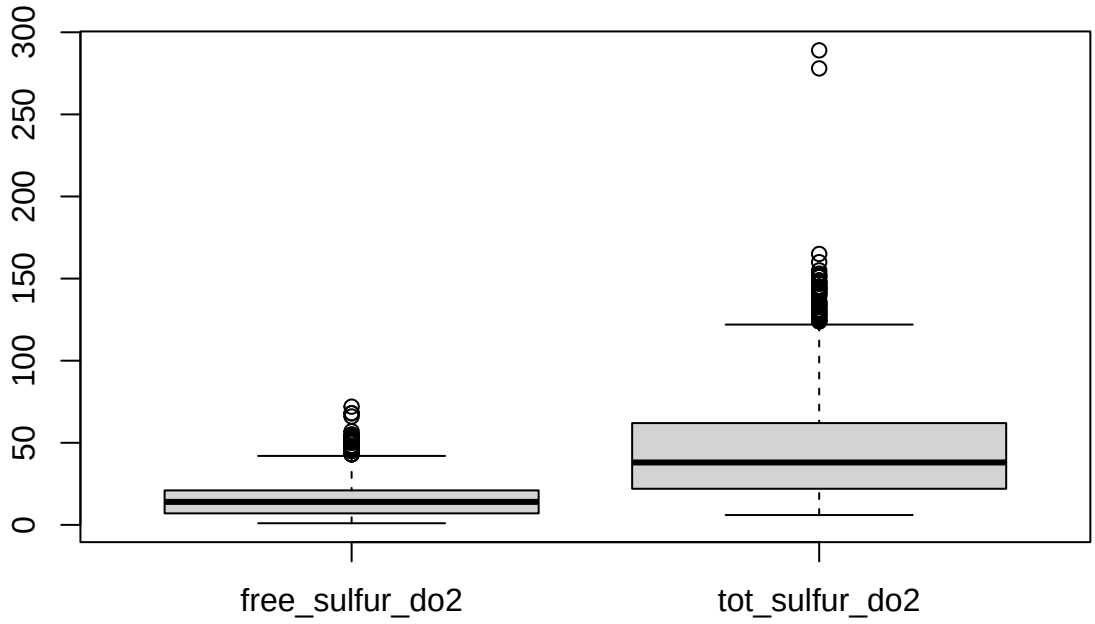
The data set do not contain missing values. However, the variable `citric_acid` contains zero values. However, ideally, we should know whether they are indeed zeroes from the client.

Outliers/Anomalies

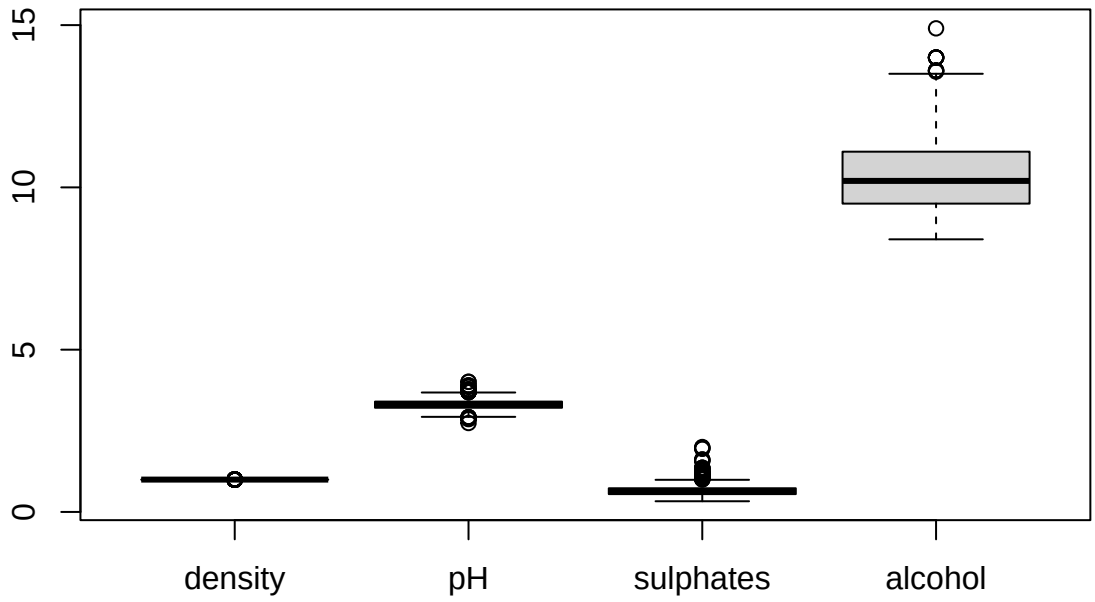
```
boxplot(red_wine[, c(1:5)])
```




```
boxplot(red_wine[, c(6:7)])
```




```
boxplot(red_wine[, c(8:11)])
```

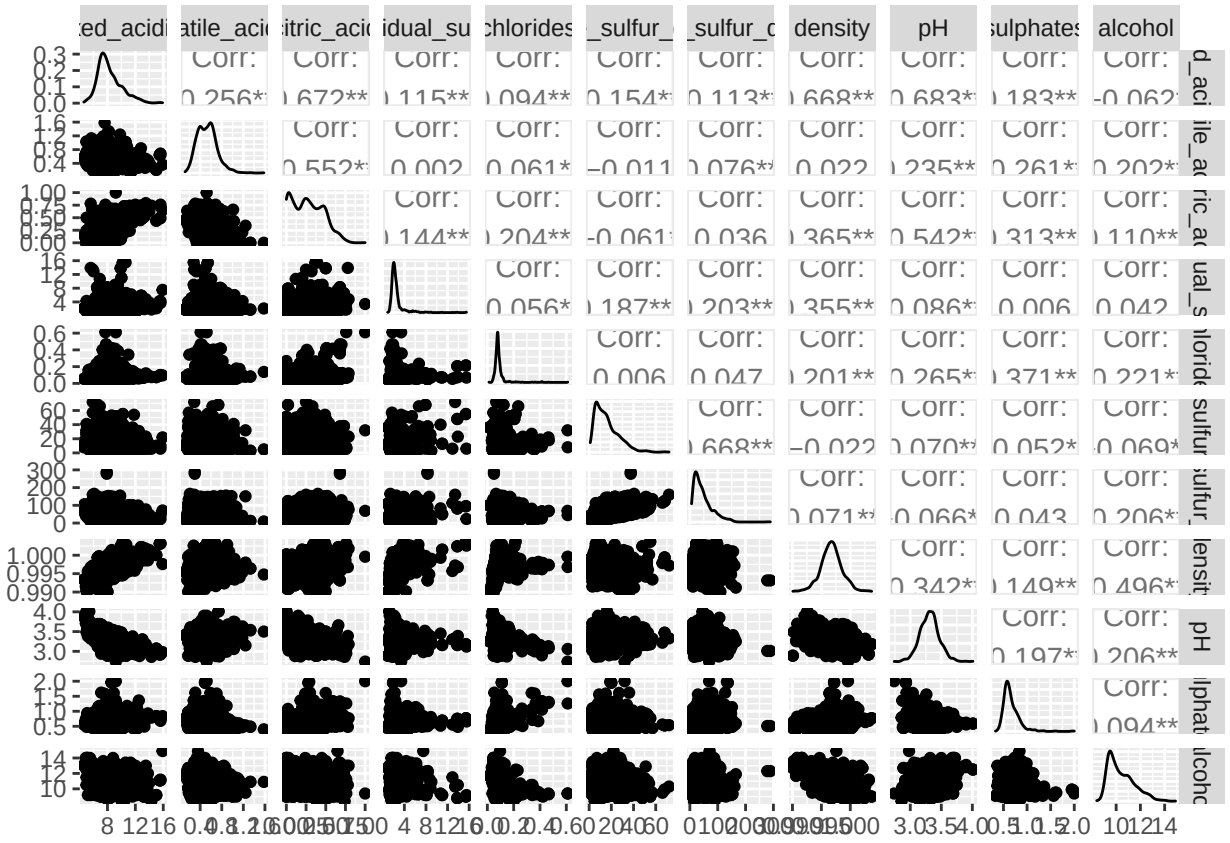



```
# install.packages("GGally")
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(red_wine,          # Data frame
        columns = 1:11) # Columns
```



The graph shows that features, including citric_acid, free_sulphur_do2, tot_sulphur_do2, and some others suffer from anomalies. Here, I use the following to cleanup some of the features using some of the preprocessing techniques. Anomalies can be problematic in accurately predicting the quality of the red wine. We will account for it by normalizing the features with Z-score normalization or standardization as it is robust to outliers/anomalies.

Near-Zero Variance Features

In this we look at the features to analyze if we have any non-variant features.

```
library(caret)
```

```
## Loading required package: lattice
```

```
nzv_features <- nearZeroVar(red_wine[, -12], names = TRUE)
print(nzv_features)
```

```
## character(0)
```

Based on the results, we do not have non-zero variant features

Highly Correlated Features

```
## Highly correlated predictors: cor__gt_90
(cor_gt_90 <- findCorrelation(cor(red_wine), names = TRUE))
```

```
## character(0)
```

```
## Highly correlated predictors (>= 98%): cor_gt_98
(cor_gt_98 <- findCorrelation(cor(red_wine), names = TRUE, cutoff = 0.98))
```

```
## character(0)
```

The features are not highly correlated.

Linear Regression

```
library(estimatr)
lr_mod1 <- lm_robust(quality ~ ., data = red_wine)
summary(lr_mod1)
```

```
##
## Call:
## lm_robust(formula = quality ~ ., data = red_wine)
##
## Standard error type:  HC2
##
## Coefficients:
##                    Estimate Std. Error t value  Pr(>|t|)   CI Lower  CI Upper
## (Intercept)       21.965208  2.447e+01  0.8975 3.696e-01 -2.604e+01 69.968344
## fixed_acidity      0.024991  3.233e-02  0.7731 4.396e-01 -3.842e-02  0.088398
## volatile_acidity  -1.083590  1.369e-01 -7.9157 4.578e-15 -1.352e+00 -0.815084
## citric_acid       -0.182564  1.527e-01 -1.1956 2.320e-01 -4.821e-01  0.116935
## residual_sugar     0.016331  1.917e-02  0.8518 3.944e-01 -2.127e-02  0.053936
## chlorides         -1.874225  4.811e-01 -3.8961 1.018e-04 -2.818e+00 -0.930651
## free_sulfur_do2    0.004361  2.242e-03  1.9452 5.193e-02 -3.647e-05  0.008759
## tot_sulfur_do2    -0.003265  7.564e-04 -4.3160 1.687e-05 -4.748e-03 -0.001781
## density          -17.881164  2.504e+01 -0.7141 4.753e-01 -6.700e+01 31.235364
## pH                -0.413653  2.135e-01 -1.9376 5.284e-02 -8.324e-01  0.005084
## sulphates          0.916334  1.346e-01  6.8056 1.422e-11  6.522e-01  1.180434
## alcohol            0.276198  2.905e-02  9.5089 6.851e-21  2.192e-01  0.333171
##                    DF
## (Intercept)      1587
## fixed_acidity    1587
## volatile_acidity 1587
## citric_acid      1587
## residual_sugar   1587
## chlorides        1587
## free_sulfur_do2  1587
## tot_sulfur_do2   1587
## density          1587
## pH               1587
```

```
## sulphates        1587
## alcohol          1587
##
## Multiple R-squared:  0.3606 ,    Adjusted R-squared:  0.3561
## F-statistic: 78.12 on 11 and 1587 DF,  p-value: < 2.2e-16
```

```
lr_mod2 <- lm_robust(quality ~ volatile_acidity + chlorides + tot_sulfur_do2 + pH + sulphates + alcohol
summary(lr_mod2)
```

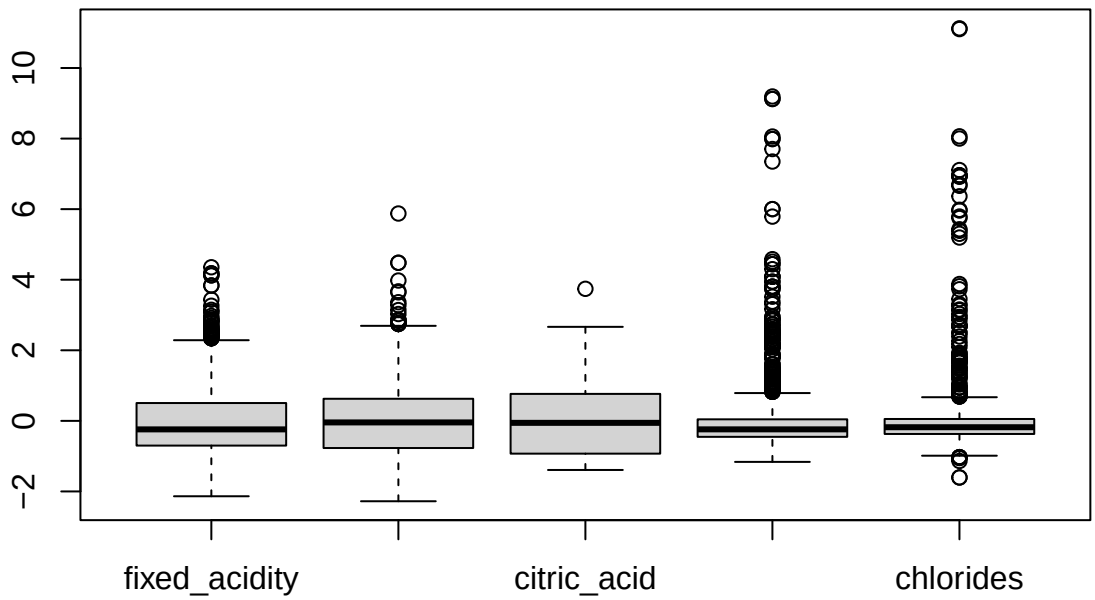
```
##
## Call:
## lm_robust(formula = quality ~ volatile_acidity + chlorides +
##     tot_sulfur_do2 + pH + sulphates + alcohol, data = red_wine)
##
## Standard error type:  HC2
##
## Coefficients:
##                   Estimate Std. Error t value  Pr(>|t|)  CI Lower  CI Upper
## (Intercept)       4.295732  0.4361275   9.850 2.931e-22  3.440287  5.151176
## volatile_acidity -1.038195  0.1147642  -9.046 4.191e-19 -1.263299 -0.813090
## chlorides        -2.002284  0.4552740  -4.398 1.165e-05 -2.895283 -1.109284
## tot_sulfur_do2   -0.002372  0.0005154  -4.603 4.505e-06 -0.003383 -0.001361
## pH               -0.435183  0.1278682  -3.403 6.821e-04 -0.685991 -0.184375
## sulphates         0.888680  0.1318932   6.738 2.239e-11  0.629978  1.147383
## alcohol           0.290674  0.0196687  14.778 2.068e-46  0.252095  0.329253
##                     DF
## (Intercept)       1592
## volatile_acidity 1592
## chlorides        1592
## tot_sulfur_do2   1592
## pH               1592
## sulphates        1592
## alcohol          1592
##
## Multiple R-squared:  0.3572 ,    Adjusted R-squared:  0.3548
## F-statistic: 144.1 on 6 and 1592 DF,  p-value: < 2.2e-16
```

As you saw from lr_mod2 with less features (after removing perceived irrelevant features) perform equally well (based on Adjusted R^2) compared with the lr_mod1 with all the features. Moving forward, we will consider only the features that are important in predicting the quality of red wine. That is, we use only the features that were included in the lr_mod2.

Data Normalization

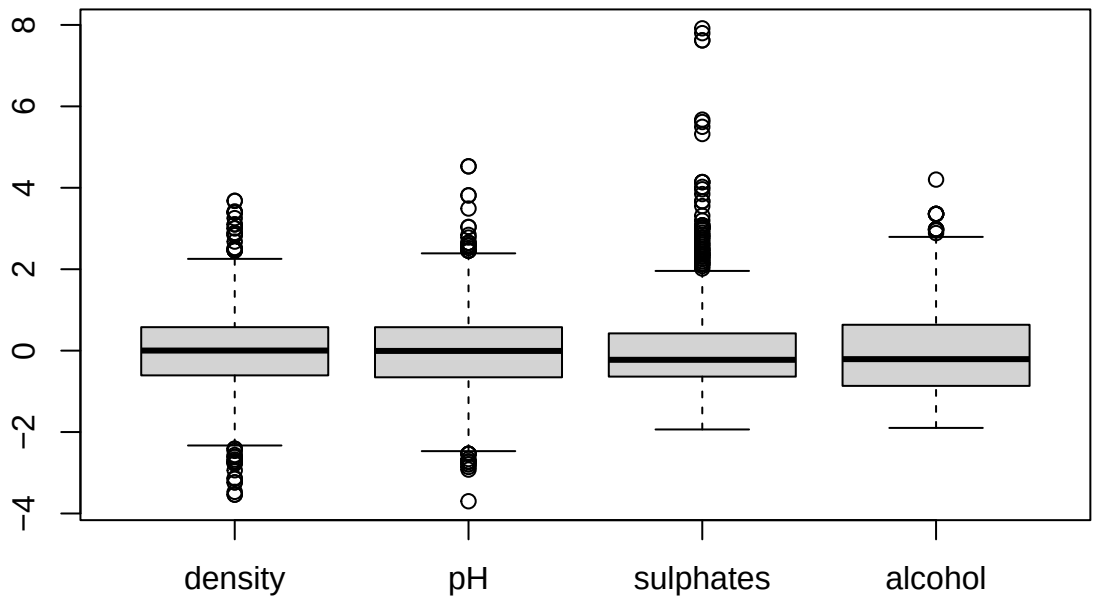
From the boxplots shown above, almost all of them has outliers at some level of degree. Therefore, we need to standardize the features.

```
red_wine_scaled <- scale(red_wine[, -12])
boxplot(red_wine_scaled[, c(1:5)])
```




```
boxplot(red_wine_scaled[, c(6:7)])
```


```
boxplot(red_wine_scaled[, c(8:11)])
```




```
all_red_wine_scaled <- data.frame(cbind(red_wine_scaled, red_wine$quality))
#str(all_red_wine_scaled)
colnames(all_red_wine_scaled)[12] <- "quality"
summary(all_red_wine_scaled)
```

```
##  fixed_acidity      volatile_acidity    citric_acid       residual_sugar
##  Min.   :-2.1364   Min.   :-2.27757   Min.   :-1.39104   Min.   :-1.1623
##  1st Qu.:-0.7005   1st Qu.:-0.76969   1st Qu.:-0.92903   1st Qu.:-0.4531
##  Median :-0.2410   Median :-0.04367   Median :-0.05634   Median :-0.2403
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.5056   3rd Qu.: 0.62649   3rd Qu.: 0.76501   3rd Qu.: 0.0434
##  Max.   : 4.3538   Max.   : 5.87614   Max.   : 3.74240   Max.   : 9.1928
##    chlorides         free_sulfur_do2   tot_sulfur_do2       density
##  Min.   :-1.60344   Min.   :-1.4221   Min.   :-1.2302   Min.   :-3.53762
##  1st Qu.:-0.37111   1st Qu.:-0.8485   1st Qu.:-0.7438   1st Qu.:-0.60757
##  Median :-0.17989   Median :-0.1792   Median :-0.2574   Median : 0.00176
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.05383   3rd Qu.: 0.4900   3rd Qu.: 0.4722   3rd Qu.: 0.57664
##  Max.   :11.12355   Max.   : 5.3656   Max.   : 7.3728   Max.   : 3.67890
##        pH              sulphates         alcohol           quality
##  Min.   :-3.69924   Min.   :-1.9359   Min.   :-1.8983   Min.   :3.000
##  1st Qu.:-0.65494   1st Qu.:-0.6380   1st Qu.:-0.8661   1st Qu.:5.000
##  Median :-0.00721   Median :-0.2251   Median :-0.2092   Median :6.000
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   :5.636
##  3rd Qu.: 0.57574   3rd Qu.: 0.4239   3rd Qu.: 0.6353   3rd Qu.:6.000
##  Max.   : 4.52687   Max.   : 7.9162   Max.   : 4.2011   Max.   :8.000
```

```
#write.csv(all_red_wine_scaled,"/Users/prithvirajlakkakula/Desktop/Regression-RedWineQuality/all_red_wi
#/Users/prithvirajlakkakula/Desktop/Regression-RedWineQuality
```

Based on the boxplots, all the features are brought to the same scale.

Exploratory Graphs

Splitting the Data into Training and Testing sets

```
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 https://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
```

```
h2o.init()
```

```
##  Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         27 minutes 47 seconds
##     H2O cluster timezone:       America/Chicago
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.36.0.2
##     H2O cluster version age:    2 months and 2 days
##     H2O cluster name:           H2O_started_from_R_prithvirajlakkakula_uyg756
```

```
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   1.38 GB
##     H2O cluster total cores:    4
##     H2O cluster allowed cores:  4
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     H2O API Extensions:         Amazon S3, XGBoost, Algos, Infogram, AutoML, Core V3, TargetEncoder,
##     R Version:                  R version 4.1.2 (2021-11-01)
```

```
df <- h2o.importFile("/Users/prithvirajlakkakula/Desktop/GitHubProjects/Regression-RedWineQuality/all_r
```

```
##   |                                                                              |
```

```
h2o.describe(df)
```

```
##               Label Type Missing Zeros PosInf NegInf       Min       Max
## 1     fixed_acidity real       0     0      0      0 -2.136377  4.353787
## 2  volatile_acidity real       0     0      0      0 -2.277567  5.876138
## 3       citric_acid real       0     0      0      0 -1.391037  3.742403
## 4    residual_sugar real       0     0      0      0 -1.162333  9.192806
## 5         chlorides real       0     0      0      0 -1.603443 11.123555
## 6   free_sulfur_do2 real       0     0      0      0 -1.422055  5.365606
## 7    tot_sulfur_do2 real       0     0      0      0 -1.230199  7.372847
## 8           density real       0     0      0      0 -3.537625  3.678904
## 9                pH real       0     0      0      0 -3.699244  4.526866
## 10        sulphates real       0     0      0      0 -1.935902  7.916200
## 11          alcohol real       0     0      0      0 -1.898325  4.201138
## 12          quality  int       0     0      0      0  3.000000  8.000000
##              Mean     Sigma Cardinality
## 1   2.488455e-16 1.0000000          NA
## 2   2.132961e-16 1.0000000          NA
## 3   7.109871e-17 1.0000000          NA
## 4  -7.998605e-17 1.0000000          NA
## 5   5.332403e-17 1.0000000          NA
## 6   0.000000e+00 1.0000000          NA
## 7   1.155354e-16 1.0000000          NA
## 8   2.420911e-14 1.0000000          NA
## 9  -5.332403e-17 1.0000000          NA
## 10 -7.109871e-17 1.0000000          NA
## 11 -5.332403e-17 1.0000000          NA
## 12  5.636023e+00 0.8075694          NA
```

```
y <- "quality"

data_splits <- h2o.splitFrame(df, ratios = 0.75, seed = 143)

training <- data_splits[[1]]
testing <- data_splits[[2]]
```

Training

```
auto_ml <- h2o.automl(y = y,
                      training_frame = training,
                      leaderboard_frame = testing,
                      max_runtime_secs = 500,
                      seed = 143)#,
```

```
##   |                                                                              |
## 13:35:56.736: Project: AutoML_4_20220328_133556
## 13:35:56.736: 5-fold cross-validation will be used.
## 13:35:56.747: Setting stopping tolerance adaptively based on the training frame: 0.02880756009578387
## 13:35:56.747: Build control seed: 143
## 13:35:56.747: training frame: Frame key: AutoML_4_20220328_133556_training_RTMP_sid_98f8_2    cols:
## 13:35:56.747: validation frame: NULL
## 13:35:56.754: leaderboard frame: Frame key: RTMP_sid_98f8_4    cols: 12    rows: 394  chunks: 16
## 13:35:56.754: blending frame: NULL
## 13:35:56.754: response column: quality
## 13:35:56.755: fold column: null
## 13:35:56.755: weights column: null
## 13:35:56.755: Loading execution steps: [{XGBoost : [def_2 (1g, 10w), def_1 (2g, 10w), def_3 (3g, 10w
## 13:35:56.756: Defined work allocations: [Work{def_2, XGBoost, ModelBuild, group=1, weight=10}, Work{
## 13:35:56.756: Actual work allocations: [Work{def_2, XGBoost, ModelBuild, group=1, weight=10}, Work{d
## 13:35:56.757: AutoML job created: 2022.03.28 13:35:56.736
## 13:35:56.757: AutoML build started: 2022.03.28 13:35:56.757
## 13:35:56.758: Time assigned for XGBoost_1_AutoML_4_20220328_133556: 142.856859375s
## 13:35:56.758: AutoML: starting XGBoost_1_AutoML_4_20220328_133556 model training
## 13:35:56.758: XGBoost_1_AutoML_4_20220328_133556 [XGBoost def_2] started
## 13:35:57.759: XGBoost_1_AutoML_4_20220328_133556 [XGBoost def_2] complete
## 13:35:57.759: Adding model XGBoost_1_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_20
## 13:35:57.762: New leader: XGBoost_1_AutoML_4_20220328_133556, mean_residual_deviance: 0.414709523641
## 13:35:57.763: Time assigned for GLM_1_AutoML_4_20220328_133556: 199.597609375s
## 13:35:57.763: AutoML: starting GLM_1_AutoML_4_20220328_133556 model training
## 13:35:57.763: GLM_1_AutoML_4_20220328_133556 [GLM def_1] started  |
## 13:35:58.764: GLM_1_AutoML_4_20220328_133556 [GLM def_1] complete
## 13:35:58.764: Adding model GLM_1_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:35:58.766: Time assigned for GBM_1_AutoML_4_20220328_133556: 331.994s
## 13:35:58.766: AutoML: starting GBM_1_AutoML_4_20220328_133556 model training
## 13:35:58.767: GBM_1_AutoML_4_20220328_133556 [GBM def_5] started  |
## 13:35:59.769: GBM_1_AutoML_4_20220328_133556 [GBM def_5] complete
## 13:35:59.769: Adding model GBM_1_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:35:59.774: Time assigned for StackedEnsemble_BestOfFamily_1_AutoML_4_20220328_133556: 496.983s
## 13:35:59.774: AutoML: starting StackedEnsemble_BestOfFamily_1_AutoML_4_20220328_133556 model trainin
## 13:35:59.774: StackedEnsemble_BestOfFamily_1_AutoML_4_20220328_133556 [StackedEnsemble best_of_famil
## 13:36:00.779: StackedEnsemble_BestOfFamily_1_AutoML_4_20220328_133556 [StackedEnsemble best_of_famil
## 13:36:00.780: Adding model StackedEnsemble_BestOfFamily_1_AutoML_4_20220328_133556 to leaderboard Lea
## 13:36:00.787: New leader: StackedEnsemble_BestOfFamily_1_AutoML_4_20220328_133556, mean_residual_devi
## 13:36:00.787: Time assigned for XGBoost_2_AutoML_4_20220328_133556: 76.303078125s
## 13:36:00.788: AutoML: starting XGBoost_2_AutoML_4_20220328_133556 model training
## 13:36:00.788: XGBoost_2_AutoML_4_20220328_133556 [XGBoost def_1] started  |
## 13:36:01.789: XGBoost_2_AutoML_4_20220328_133556 [XGBoost def_1] complete
## 13:36:01.789: Adding model XGBoost_2_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_20
## 13:36:01.792: Time assigned for DRF_1_AutoML_4_20220328_133556: 89.993640625s
```

```
## 13:36:01.792: AutoML: starting DRF_1_AutoML_4_20220328_133556 model training
## 13:36:01.793: DRF_1_AutoML_4_20220328_133556 [DRF def_1] started  |
## 13:36:03.795: DRF_1_AutoML_4_20220328_133556 [DRF def_1] complete
## 13:36:03.795: Adding model DRF_1_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:36:03.801: New leader: DRF_1_AutoML_4_20220328_133556, mean_residual_deviance: 0.3711118793922384
## 13:36:03.802: Time assigned for GBM_2_AutoML_4_20220328_133556: 109.5455546875s
## 13:36:03.802: AutoML: starting GBM_2_AutoML_4_20220328_133556 model training
## 13:36:03.802: GBM_2_AutoML_4_20220328_133556 [GBM def_2] started
## 13:36:04.804: GBM_2_AutoML_4_20220328_133556 [GBM def_2] complete
## 13:36:04.804: Adding model GBM_2_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:36:04.809: Time assigned for GBM_3_AutoML_4_20220328_133556: 140.556578125s
## 13:36:04.809: AutoML: starting GBM_3_AutoML_4_20220328_133556 model training
## 13:36:04.809: GBM_3_AutoML_4_20220328_133556 [GBM def_3] started  |
## 13:36:05.814: GBM_3_AutoML_4_20220328_133556 [GBM def_3] complete
## 13:36:05.814: Adding model GBM_3_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:36:05.819: Time assigned for GBM_4_AutoML_4_20220328_133556: 196.375203125s
## 13:36:05.819: AutoML: starting GBM_4_AutoML_4_20220328_133556 model training
## 13:36:05.819: GBM_4_AutoML_4_20220328_133556 [GBM def_4] started  |
## 13:36:06.823: GBM_4_AutoML_4_20220328_133556 [GBM def_4] complete
## 13:36:06.823: Adding model GBM_4_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:36:06.830: Time assigned for StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556: 163.309s
## 13:36:06.830: AutoML: starting StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556 model training
## 13:36:06.830: StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:36:07.836: StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:36:07.836: Adding model StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556 to leaderboard Lea
## 13:36:07.847: New leader: StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556, mean_residual_devi
## 13:36:07.847: Time assigned for StackedEnsemble_AllModels_1_AutoML_4_20220328_133556: 488.91s
## 13:36:07.847: AutoML: starting StackedEnsemble_AllModels_1_AutoML_4_20220328_133556 model training
## 13:36:07.847: StackedEnsemble_AllModels_1_AutoML_4_20220328_133556 [StackedEnsemble all_2 (built with
## 13:36:08.848: StackedEnsemble_AllModels_1_AutoML_4_20220328_133556 [StackedEnsemble all_2 (built with
## 13:36:08.848: Adding model StackedEnsemble_AllModels_1_AutoML_4_20220328_133556 to leaderboard Leade
## 13:36:08.862: Time assigned for XGBoost_3_AutoML_4_20220328_133556: 88.7081875s
## 13:36:08.862: AutoML: starting XGBoost_3_AutoML_4_20220328_133556 model training
## 13:36:08.863: XGBoost_3_AutoML_4_20220328_133556 [XGBoost def_3] started  |
## 13:36:09.865: XGBoost_3_AutoML_4_20220328_133556 [XGBoost def_3] complete
## 13:36:09.865: Adding model XGBoost_3_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_20
## 13:36:09.868: Time assigned for XRT_1_AutoML_4_20220328_133556: 108.1975546875s
## 13:36:09.868: AutoML: starting XRT_1_AutoML_4_20220328_133556 model training
## 13:36:09.869: XRT_1_AutoML_4_20220328_133556 [DRF XRT (Extremely Randomized Trees)] started  |
## 13:36:11.870: XRT_1_AutoML_4_20220328_133556 [DRF XRT (Extremely Randomized Trees)] complete
## 13:36:11.870: Adding model XRT_1_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:36:11.877: Time assigned for GBM_5_AutoML_4_20220328_133556: 138.53715625s
## 13:36:11.877: AutoML: starting GBM_5_AutoML_4_20220328_133556 model training
## 13:36:11.877: GBM_5_AutoML_4_20220328_133556 [GBM def_1] started  |
## 13:36:12.882: GBM_5_AutoML_4_20220328_133556 [GBM def_1] complete
## 13:36:12.882: Adding model GBM_5_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML_4_202203
## 13:36:12.888: Time assigned for DeepLearning_1_AutoML_4_20220328_133556: 193.547609375s
## 13:36:12.888: AutoML: starting DeepLearning_1_AutoML_4_20220328_133556 model training
## 13:36:12.888: DeepLearning_1_AutoML_4_20220328_133556 [DeepLearning def_1] started  |
## 13:36:13.890: DeepLearning_1_AutoML_4_20220328_133556 [DeepLearning def_1] complete
## 13:36:13.890: Adding model DeepLearning_1_AutoML_4_20220328_133556 to leaderboard Leaderboard_AutoML
## 13:36:13.895: Time assigned for StackedEnsemble_BestOfFamily_3_AutoML_4_20220328_133556: 160.954s
## 13:36:13.895: AutoML: starting StackedEnsemble_BestOfFamily_3_AutoML_4_20220328_133556 model training
## 13:36:13.895: StackedEnsemble_BestOfFamily_3_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
```

```
## 13:36:14.898: StackedEnsemble_BestOfFamily_3_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:36:14.898: Adding model StackedEnsemble_BestOfFamily_3_AutoML_4_20220328_133556 to leaderboard Lea
## 13:36:14.913: Time assigned for StackedEnsemble_AllModels_2_AutoML_4_20220328_133556: 481.844s
## 13:36:14.913: AutoML: starting StackedEnsemble_AllModels_2_AutoML_4_20220328_133556 model training
## 13:36:14.914: StackedEnsemble_AllModels_2_AutoML_4_20220328_133556 [StackedEnsemble all_3 (built with
## 13:36:15.915: StackedEnsemble_AllModels_2_AutoML_4_20220328_133556 [StackedEnsemble all_3 (built with
## 13:36:15.915: Adding model StackedEnsemble_AllModels_2_AutoML_4_20220328_133556 to leaderboard Leader
## 13:36:15.934: Time assigned for XGBoost_grid_1_AutoML_4_20220328_133556: 221.9183125s
## 13:36:15.934: AutoML: starting XGBoost_grid_1_AutoML_4_20220328_133556 hyperparameter search
## 13:36:15.934: XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost Grid Search] started  |
## 13:36:16.935: Built: 1 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:16.935: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_1 to leaderboard Leaderboard
## 13:36:17.939: Built: 3 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:17.939: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_3 to leaderboard Leaderboard
## 13:36:17.939: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_2 to leaderboard Leaderboard
## 13:36:18.962: Built: 5 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:18.962: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_4 to leaderboard Leaderboard
## 13:36:18.962: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_5 to leaderboard Leaderboard
## 13:36:20.8: Built: 7 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost G
## 13:36:20.8: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_6 to leaderboard Leaderboard_A
## 13:36:20.8: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_7 to leaderboard Leaderboard_A
## 13:36:21.12: Built: 9 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:21.12: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_8 to leaderboard Leaderboard_
## 13:36:21.12: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_9 to leaderboard Leaderboard_
## 13:36:22.19: Built: 11 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:22.19: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_11 to leaderboard Leaderboard
## 13:36:22.19: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_10 to leaderboard Leaderboard
## 13:36:23.25: Built: 14 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:23.25: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_14 to leaderboard Leaderboard
## 13:36:23.25: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_13 to leaderboard Leaderboard
## 13:36:23.25: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_12 to leaderboard Leaderboard
## 13:36:24.42: Built: 16 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:24.42: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_16 to leaderboard Leaderboard
## 13:36:24.42: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_15 to leaderboard Leaderboard
## 13:36:25.60: Built: 19 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:25.60: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_19 to leaderboard Leaderboard
## 13:36:25.60: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_18 to leaderboard Leaderboard
## 13:36:25.60: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_17 to leaderboard Leaderboard
## 13:36:26.73: Built: 21 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:26.73: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_21 to leaderboard Leaderboard
## 13:36:26.73: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_20 to leaderboard Leaderboard
## 13:36:27.94: Built: 23 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:27.94: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_23 to leaderboard Leaderboard
## 13:36:27.94: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_22 to leaderboard Leaderboard
## 13:36:27.99: New leader: XGBoost_grid_1_AutoML_4_20220328_133556_model_23, mean_residual_deviance: 0
## 13:36:28.99: Built: 24 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost
## 13:36:28.100: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_24 to leaderboard Leaderboar
## 13:36:29.105: Built: 26 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:29.105: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_26 to leaderboard Leaderboar
## 13:36:29.106: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_25 to leaderboard Leaderboar
## 13:36:30.121: Built: 28 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:30.121: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_27 to leaderboard Leaderboar
## 13:36:30.121: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_28 to leaderboard Leaderboar
## 13:36:31.128: Built: 30 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
```

```
## 13:36:31.128: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_30 to leaderboard Leaderboar
## 13:36:31.128: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_29 to leaderboard Leaderboar
## 13:36:32.134: Built: 32 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:32.135: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_31 to leaderboard Leaderboar
## 13:36:32.135: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_32 to leaderboard Leaderboar
## 13:36:33.139: Built: 33 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:33.140: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_33 to leaderboard Leaderboar
## 13:36:34.153: Built: 35 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:34.153: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_35 to leaderboard Leaderboar
## 13:36:34.153: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_34 to leaderboard Leaderboar
## 13:36:35.160: Built: 37 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:35.160: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_36 to leaderboard Leaderboar
## 13:36:35.160: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_37 to leaderboard Leaderboar
## 13:36:36.180: Built: 39 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:36.180: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_38 to leaderboard Leaderboar
## 13:36:36.180: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_39 to leaderboard Leaderboar
## 13:36:37.193: Built: 42 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:37.193: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_40 to leaderboard Leaderboar
## 13:36:37.193: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_42 to leaderboard Leaderboar
## 13:36:37.193: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_41 to leaderboard Leaderboar
## 13:36:38.213: Built: 44 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:38.213: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_44 to leaderboard Leaderboar
## 13:36:38.213: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_43 to leaderboard Leaderboar
## 13:36:39.244: Built: 46 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:39.244: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_46 to leaderboard Leaderboar
## 13:36:39.244: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_45 to leaderboard Leaderboar
## 13:36:40.256: Built: 48 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:40.256: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_48 to leaderboard Leaderboar
## 13:36:40.256: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_47 to leaderboard Leaderboar
## 13:36:41.269: Built: 50 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:41.270: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_50 to leaderboard Leaderboar
## 13:36:41.270: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_49 to leaderboard Leaderboar
## 13:36:42.280: Built: 52 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:42.280: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_52 to leaderboard Leaderboar
## 13:36:42.280: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_51 to leaderboard Leaderboar
## 13:36:43.287: Built: 54 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:43.288: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_54 to leaderboard Leaderboar
## 13:36:43.288: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_53 to leaderboard Leaderboar
## 13:36:44.302: Built: 56 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:44.302: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_56 to leaderboard Leaderboar
## 13:36:44.302: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_55 to leaderboard Leaderboar
## 13:36:45.308: Built: 59 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:45.308: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_58 to leaderboard Leaderboar
## 13:36:45.308: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_59 to leaderboard Leaderboar
## 13:36:45.308: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_57 to leaderboard Leaderboar
## 13:36:46.332: Built: 60 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:46.333: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_60 to leaderboard Leaderboar
## 13:36:47.338: Built: 63 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:47.339: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_63 to leaderboard Leaderboar
## 13:36:47.339: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_61 to leaderboard Leaderboar
## 13:36:47.339: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_62 to leaderboard Leaderboar
## 13:36:48.356: Built: 66 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoos
## 13:36:48.356: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_65 to leaderboard Leaderboar
## 13:36:48.356: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_64 to leaderboard Leaderboar
```

```
## 13:36:48.356: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_66 to leaderboard Leaderboa
## 13:36:49.365: Built: 67 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoo
## 13:36:49.365: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_67 to leaderboard Leaderboa
## 13:36:50.376: Built: 69 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoo
## 13:36:50.376: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_68 to leaderboard Leaderboa
## 13:36:50.376: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_69 to leaderboard Leaderboa
## 13:36:51.414: XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoost Grid Search] complete
## 13:36:51.414: Built: 70 models for HyperparamSearch : XGBoost_grid_1_AutoML_4_20220328_133556 [XGBoo
## 13:36:51.415: Adding model XGBoost_grid_1_AutoML_4_20220328_133556_model_70 to leaderboard Leaderboa
## 13:36:51.424: Time assigned for GBM_grid_1_AutoML_4_20220328_133556: 254.476015625s
## 13:36:51.424: AutoML: starting GBM_grid_1_AutoML_4_20220328_133556 hyperparameter search
## 13:36:51.424: GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Search] started
## 13:36:52.429: Built: 1 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Se
## 13:36:52.429: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_1 to leaderboard Leaderboard_Aut
## 13:36:53.445: Built: 3 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Se
## 13:36:53.445: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_3 to leaderboard Leaderboard_Aut
## 13:36:53.445: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_2 to leaderboard Leaderboard_Aut
## 13:36:54.470: Built: 4 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Se
## 13:36:54.470: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_4 to leaderboard Leaderboard_Aut
## 13:36:55.482: Built: 5 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Se
## 13:36:55.482: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_5 to leaderboard Leaderboard_Aut
## 13:36:56.493: Built: 7 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Se
## 13:36:56.493: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_6 to leaderboard Leaderboard_Aut
## 13:36:56.493: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_7 to leaderboard Leaderboard_Aut
## 13:36:57.510: Built: 8 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Se
## 13:36:57.510: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_8 to leaderboard Leaderboard_Aut
## 13:36:58.521: Built: 10 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid S
## 13:36:58.521: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_10 to leaderboard Leaderboard_Au
## 13:36:58.521: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_9 to leaderboard Leaderboard_Aut
## 13:36:59.533: Built: 12 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid S
## 13:36:59.533: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_12 to leaderboard Leaderboard_Au
## 13:36:59.533: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_11 to leaderboard Leaderboard_Au
## 13:37:00.552: Built: 13 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid S
## 13:37:00.552: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_13 to leaderboard Leaderboard_Au
## 13:37:01.570: Built: 14 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid S
## 13:37:01.570: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_14 to leaderboard Leaderboard_Au
## 13:37:02.584: Built: 15 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid S
## 13:37:02.584: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_15 to leaderboard Leaderboard_Au
## 13:37:03.598: GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid Search] complete
## 13:37:03.598: Built: 16 models for HyperparamSearch : GBM_grid_1_AutoML_4_20220328_133556 [GBM Grid S
## 13:37:03.599: Adding model GBM_grid_1_AutoML_4_20220328_133556_model_16 to leaderboard Leaderboard_Au
## 13:37:03.617: Time assigned for DeepLearning_grid_1_AutoML_4_20220328_133556: 288.7606875s
## 13:37:03.617: AutoML: starting DeepLearning_grid_1_AutoML_4_20220328_133556 hyperparameter search
## 13:37:03.618: DeepLearning_grid_1_AutoML_4_20220328_133556 [DeepLearning Grid Search] started  |
## 13:38:13.829: Built: 1 models for HyperparamSearch : DeepLearning_grid_1_AutoML_4_20220328_133556 [De
## 13:38:13.829: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_1 to leaderboard Leader
## 13:39:18.39: Built: 2 models for HyperparamSearch : DeepLearning_grid_1_AutoML_4_20220328_133556 [Dee
## 13:39:18.39: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_2 to leaderboard Leaderb
## 13:40:24.278: Built: 3 models for HyperparamSearch : DeepLearning_grid_1_AutoML_4_20220328_133556 [De
## 13:40:24.278: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_3 to leaderboard Leader
## 13:41:32.537: Built: 4 models for HyperparamSearch : DeepLearning_grid_1_AutoML_4_20220328_133556 [De
## 13:41:32.537: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_4 to leaderboard Leader
## 13:41:48.615: Built: 5 models for HyperparamSearch : DeepLearning_grid_1_AutoML_4_20220328_133556 [De
## 13:41:48.615: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_5 to leaderboard Leader
```

```
## 13:41:51.644: Built: 6 models for HyperparamSearch : DeepLearning_grid_1_AutoML_4_20220328_133556 [De
## 13:41:51.644: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_6 to leaderboard Leade
## 13:41:52.686: DeepLearning_grid_1_AutoML_4_20220328_133556 [DeepLearning Grid Search] complete
## 13:41:52.686: Built: 8 models for HyperparamSearch : DeepLearning_grid_1_AutoML_4_20220328_133556 [De
## 13:41:52.686: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_8 to leaderboard Leade
## 13:41:52.686: Adding model DeepLearning_grid_1_AutoML_4_20220328_133556_model_7 to leaderboard Leade
## 13:41:52.699: Time assigned for StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556: 48.01933593
## 13:41:52.699: AutoML: starting StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556 model trainin
## 13:41:52.700: StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:41:53.702: StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:41:53.703: Adding model StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556 to leaderboard Lea
## 13:41:53.726: New leader: StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556, mean_residual_devi
## 13:41:53.727: Time assigned for StackedEnsemble_AllModels_3_AutoML_4_20220328_133556: 143.03s
## 13:41:53.727: AutoML: starting StackedEnsemble_AllModels_3_AutoML_4_20220328_133556 model training
## 13:41:53.729: StackedEnsemble_AllModels_3_AutoML_4_20220328_133556 [StackedEnsemble all_4 (built with
## 13:41:54.731: StackedEnsemble_AllModels_3_AutoML_4_20220328_133556 [StackedEnsemble all_4 (built with
## 13:41:54.731: Adding model StackedEnsemble_AllModels_3_AutoML_4_20220328_133556 to leaderboard Leade
## 13:41:54.789: New leader: StackedEnsemble_AllModels_3_AutoML_4_20220328_133556, mean_residual_deviano
## 13:41:54.790: Time assigned for DeepLearning_grid_2_AutoML_4_20220328_133556: 56.78680078125s
## 13:41:54.790: AutoML: starting DeepLearning_grid_2_AutoML_4_20220328_133556 hyperparameter search
## 13:41:54.790: DeepLearning_grid_2_AutoML_4_20220328_133556 [DeepLearning Grid Search] started  |
## 13:42:40.926: Built: 1 models for HyperparamSearch : DeepLearning_grid_2_AutoML_4_20220328_133556 [De
## 13:42:40.926: Adding model DeepLearning_grid_2_AutoML_4_20220328_133556_model_1 to leaderboard Leade
## 13:42:48.992: Built: 2 models for HyperparamSearch : DeepLearning_grid_2_AutoML_4_20220328_133556 [De
## 13:42:48.993: Adding model DeepLearning_grid_2_AutoML_4_20220328_133556_model_2 to leaderboard Leade
## 13:42:53.37: DeepLearning_grid_2_AutoML_4_20220328_133556 [DeepLearning Grid Search] complete
## 13:42:53.37: Built: 3 models for HyperparamSearch : DeepLearning_grid_2_AutoML_4_20220328_133556 [Dee
## 13:42:53.37: Adding model DeepLearning_grid_2_AutoML_4_20220328_133556_model_3 to leaderboard Leaderb
## 13:42:53.56: Time assigned for DeepLearning_grid_3_AutoML_4_20220328_133556: 55.8013359375s
## 13:42:53.56: AutoML: starting DeepLearning_grid_3_AutoML_4_20220328_133556 hyperparameter search
## 13:42:53.56: DeepLearning_grid_3_AutoML_4_20220328_133556 [DeepLearning Grid Search] started  |
## 13:43:40.192: Built: 1 models for HyperparamSearch : DeepLearning_grid_3_AutoML_4_20220328_133556 [De
## 13:43:40.192: Adding model DeepLearning_grid_3_AutoML_4_20220328_133556_model_1 to leaderboard Leade
## 13:43:47.233: Built: 2 models for HyperparamSearch : DeepLearning_grid_3_AutoML_4_20220328_133556 [De
## 13:43:47.233: Adding model DeepLearning_grid_3_AutoML_4_20220328_133556_model_2 to leaderboard Leade
## 13:43:52.282: DeepLearning_grid_3_AutoML_4_20220328_133556 [DeepLearning Grid Search] complete
## 13:43:52.282: Built: 3 models for HyperparamSearch : DeepLearning_grid_3_AutoML_4_20220328_133556 [De
## 13:43:52.282: Adding model DeepLearning_grid_3_AutoML_4_20220328_133556_model_3 to leaderboard Leade
## 13:43:52.304: Time assigned for StackedEnsemble_AllModels_4_AutoML_4_20220328_133556: 24.453s
## 13:43:52.305: AutoML: starting StackedEnsemble_AllModels_4_AutoML_4_20220328_133556 model training
## 13:43:52.305: StackedEnsemble_AllModels_4_AutoML_4_20220328_133556 [StackedEnsemble all_5 (built with
## 13:43:53.309: StackedEnsemble_AllModels_4_AutoML_4_20220328_133556 [StackedEnsemble all_5 (built with
## 13:43:53.309: Adding model StackedEnsemble_AllModels_4_AutoML_4_20220328_133556 to leaderboard Leade
## 13:43:53.366: Time assigned for XGBoost_lr_search_selection_AutoML_4_20220328_133556: 3.693315673828
## 13:43:53.366: XGBoost_lr_search_selection_AutoML_4_20220328_133556 [XGBoost lr_search] started
## 13:43:53.366: Applying learning rate search on best XGBoost: XGBoost_grid_1_AutoML_4_20220328_133556
## 13:43:53.366: AutoML: starting XGBoost_lr_search_selection_AutoML_4_20220328_133556_select model trai
## 13:43:59.387: XGBoost_lr_search_selection_AutoML_4_20220328_133556 [XGBoost lr_search] complete
## 13:43:59.388: Time assigned for GBM_lr_annealing_selection_AutoML_4_20220328_133556: 1.0855625s
## 13:43:59.389: GBM_lr_annealing_selection_AutoML_4_20220328_133556 [GBM lr_annealing] started
## 13:43:59.389: Retraining best GBM with learning rate annealing: GBM_4_AutoML_4_20220328_133556
## 13:43:59.389: AutoML: starting GBM_lr_annealing_selection_AutoML_4_20220328_133556_select_model model
## 13:44:02.399: GBM_lr_annealing_selection_AutoML_4_20220328_133556 [GBM lr_annealing] complete
## 13:44:02.402: No base models, due to timeouts or the exclude_algos option. Skipping StackedEnsemble
```

```
## 13:44:02.403: Time assigned for StackedEnsemble_BestOfFamily_5_AutoML_4_20220328_133556: 7.177s
## 13:44:02.403: AutoML: starting StackedEnsemble_BestOfFamily_5_AutoML_4_20220328_133556 model training
## 13:44:02.404: StackedEnsemble_BestOfFamily_5_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:44:04.409: StackedEnsemble_BestOfFamily_5_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:44:04.409: Adding model StackedEnsemble_BestOfFamily_5_AutoML_4_20220328_133556 to leaderboard Lea
## 13:44:04.438: Time assigned for StackedEnsemble_BestOfFamily_6_AutoML_4_20220328_133556: 12.319s
## 13:44:04.438: AutoML: starting StackedEnsemble_BestOfFamily_6_AutoML_4_20220328_133556 model training
## 13:44:04.439: StackedEnsemble_BestOfFamily_6_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:44:05.440: StackedEnsemble_BestOfFamily_6_AutoML_4_20220328_133556 [StackedEnsemble best_of_family
## 13:44:05.441: Adding model StackedEnsemble_BestOfFamily_6_AutoML_4_20220328_133556 to leaderboard Lea
## 13:44:05.465: Time assigned for StackedEnsemble_AllModels_5_AutoML_4_20220328_133556: 5.646s
## 13:44:05.465: AutoML: starting StackedEnsemble_AllModels_5_AutoML_4_20220328_133556 model training
## 13:44:05.466: StackedEnsemble_AllModels_5_AutoML_4_20220328_133556 [StackedEnsemble all_xgboost (buil
## 13:44:11.481: StackedEnsemble_AllModels_5_AutoML_4_20220328_133556 [StackedEnsemble all_xgboost (buil
## 13:44:11.481: Adding model StackedEnsemble_AllModels_5_AutoML_4_20220328_133556 to leaderboard Leader
## 13:44:11.619: Time assigned for StackedEnsemble_AllModels_6_AutoML_4_20220328_133556: 5.138s
## 13:44:11.619: AutoML: starting StackedEnsemble_AllModels_6_AutoML_4_20220328_133556 model training
## 13:44:11.620: StackedEnsemble_AllModels_6_AutoML_4_20220328_133556 [StackedEnsemble all_gbm (built wi
## 13:44:17.626: StackedEnsemble_AllModels_6_AutoML_4_20220328_133556 [StackedEnsemble all_gbm (built wi
## 13:44:17.626: Adding model StackedEnsemble_AllModels_6_AutoML_4_20220328_133556 to leaderboard Leader
## 13:44:17.788: AutoML: out of time; skipping StackedEnsemble best_of_family_xglm (built with AUTO meta
## 13:44:17.788: AutoML: out of time; skipping StackedEnsemble all_xglm (built with AUTO metalearner, us
## 13:44:17.788: AutoML: out of time; skipping completion resume_best_grids
## 13:44:17.788: AutoML: out of time; skipping StackedEnsemble best_of_family (built with AUTO metalearn
## 13:44:17.788: AutoML: out of time; skipping StackedEnsemble best_N (built with AUTO metalearner, usin
## 13:44:17.788: Actual modeling steps: [{XGBoost : [def_2 (1g, 10w)]}, {GLM : [def_1 (1g, 10w)]}, {GBM
## 13:44:17.788: AutoML build stopped: 2022.03.28 13:44:17.788
## 13:44:17.788: AutoML build done: built 112 models
## 13:44:17.788: AutoML duration:  8 min 21.031 sec
## 13:44:17.811: Verifying training frame immutability. . .
## 13:44:17.811: Training frame was not mutated (as expected).
```

```
                  #project_name = "winequality_lb_frame")
```

Predicting on test set

```
print(auto_ml@leaderboard)
```

```
##                                                 model_id
## 1    StackedEnsemble_AllModels_3_AutoML_4_20220328_133556
## 2    StackedEnsemble_AllModels_4_AutoML_4_20220328_133556
## 3 StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556
## 4 StackedEnsemble_BestOfFamily_6_AutoML_4_20220328_133556
## 5        XGBoost_grid_1_AutoML_4_20220328_133556_model_23
## 6 StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556
##   mean_residual_deviance      rmse       mse       mae      rmsle
## 1              0.3436855 0.5862469 0.3436855 0.4234603 0.09521005
## 2              0.3436855 0.5862469 0.3436855 0.4234603 0.09521005
## 3              0.3470575 0.5891158 0.3470575 0.4183246 0.09542250
## 4              0.3575176 0.5979277 0.3575176 0.4185890 0.09687578
## 5              0.3586580 0.5988806 0.3586580 0.4128412 0.09667850
```

```
## 6              0.3606941 0.6005782 0.3606941 0.4294703 0.09711357
##
## [124 rows x 6 columns]
```

```
preds <- h2o.predict(auto_ml, testing)
```

```
##   |                                                                              |
```

```
head(preds)
```

```
##    predict
## 1 5.279998
## 2 5.685546
## 3 5.251997
## 4 5.512965
## 5 5.093294
## 6 5.341936
```

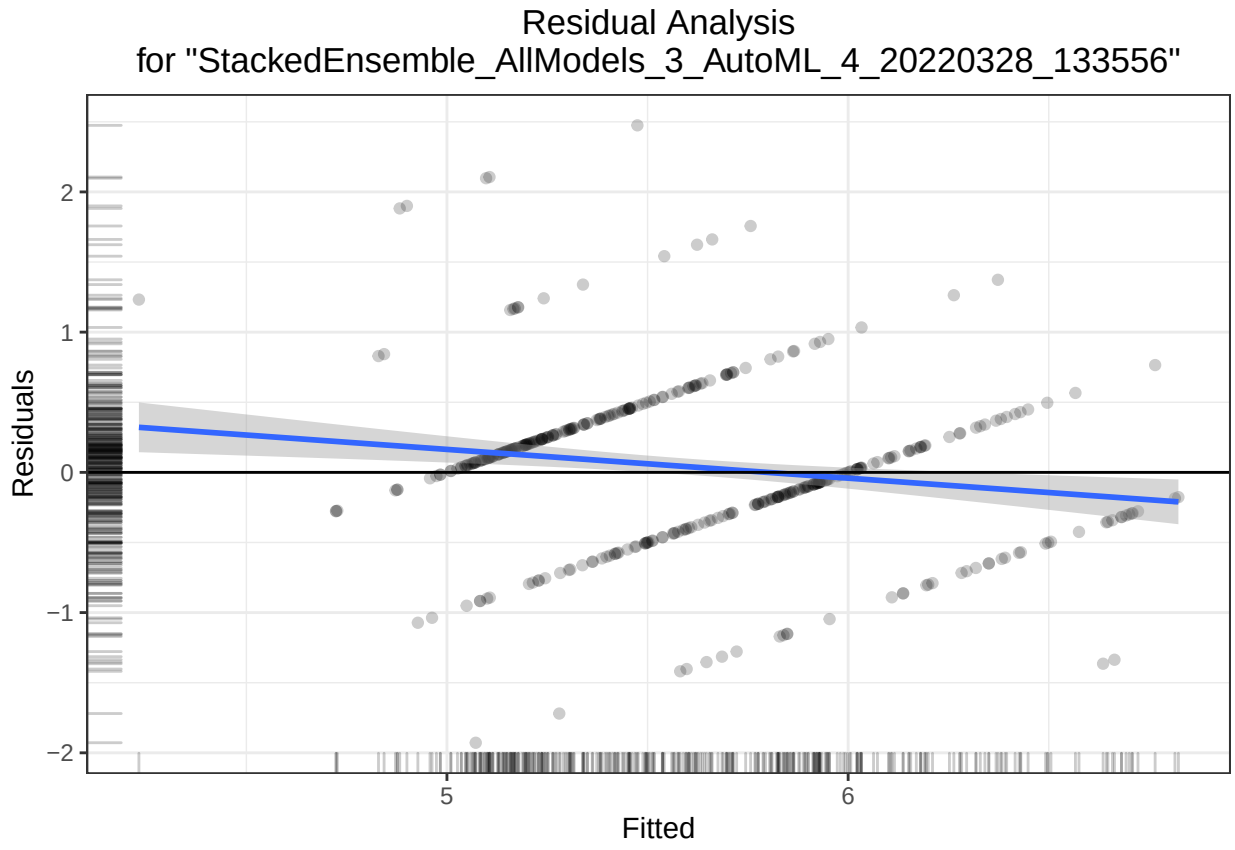
```
aml_perf <- h2o.performance(auto_ml@leader, testing)
aml_perf
```

```
## H2ORegressionMetrics: stackedensemble
##
## MSE:  0.3436855
## RMSE:  0.5862469
## MAE:  0.4234603
## RMSLE:  0.09521005
## Mean Residual Deviance :  0.3436855
```

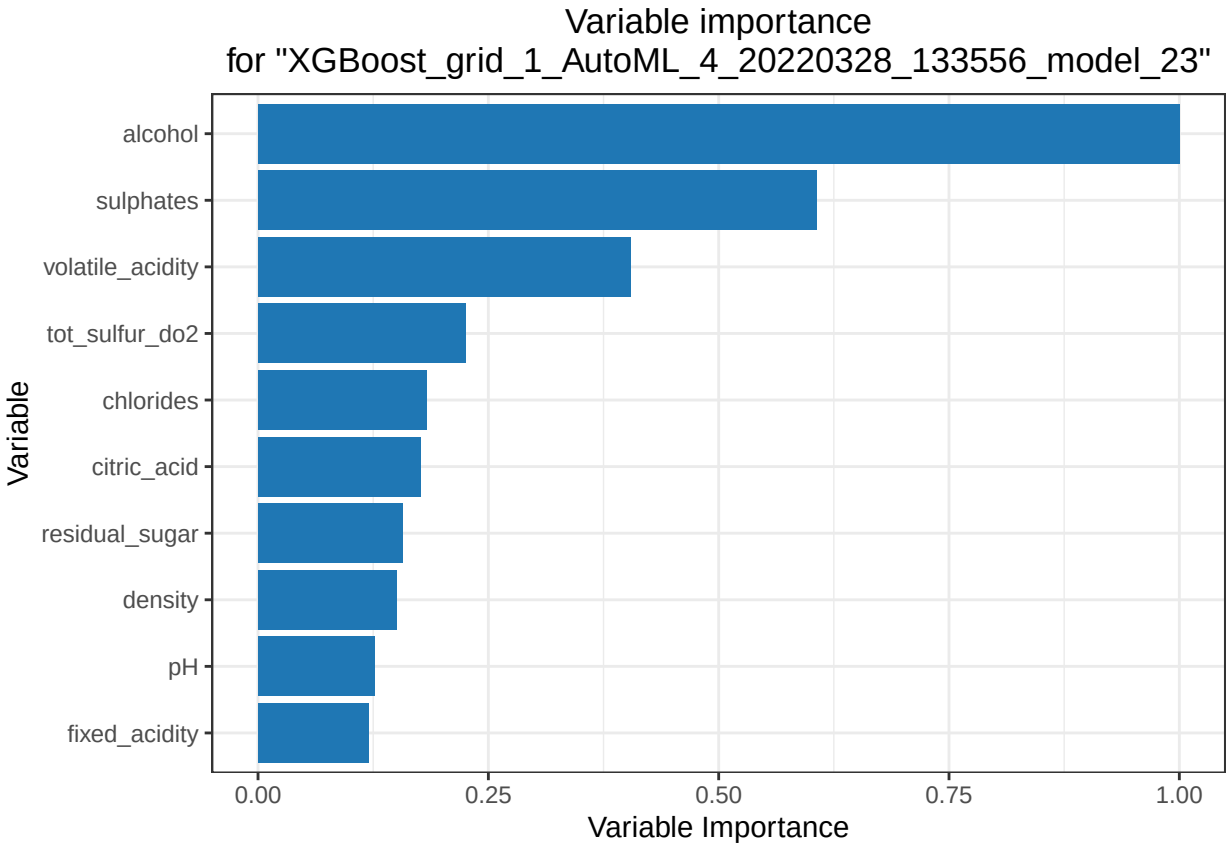
```
explns <- h2o.explain(auto_ml, testing)
explns
```

```
##
##
## Leaderboard
## ===========
##
## > Leaderboard shows models with their metrics. When provided with H2OAutoML object, the leaderboard s
##
##
## |  | model_id | mean_residual_deviance | rmse | mse | mae | rmsle | training_time_ms | predict_time_p
## |:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|
## | **1** |StackedEnsemble_AllModels_3_AutoML_4_20220328_133556 | 0.343685470184656 | 0.586246936183599
## | **2** |StackedEnsemble_AllModels_4_AutoML_4_20220328_133556 | 0.343685470184656 | 0.586246936183599
## | **3** |StackedEnsemble_BestOfFamily_4_AutoML_4_20220328_133556 | 0.347057466886373 | 0.589115834863
## | **4** |StackedEnsemble_BestOfFamily_6_AutoML_4_20220328_133556 | 0.357517561313973 | 0.597927722483
## | **5** |XGBoost_grid_1_AutoML_4_20220328_133556_model_23 | 0.358657982765416 | 0.598880608106003 | 0
## | **6** |StackedEnsemble_BestOfFamily_2_AutoML_4_20220328_133556 | 0.36069413184605 | 0.600578164643
## | **7** |StackedEnsemble_AllModels_1_AutoML_4_20220328_133556 | 0.361085245852051 | 0.60090369099553
## | **8** |StackedEnsemble_AllModels_6_AutoML_4_20220328_133556 | 0.361433305884099 | 0.601193235061822
## | **9** |StackedEnsemble_BestOfFamily_3_AutoML_4_20220328_133556 | 0.362505725168391 | 0.602084483414
```

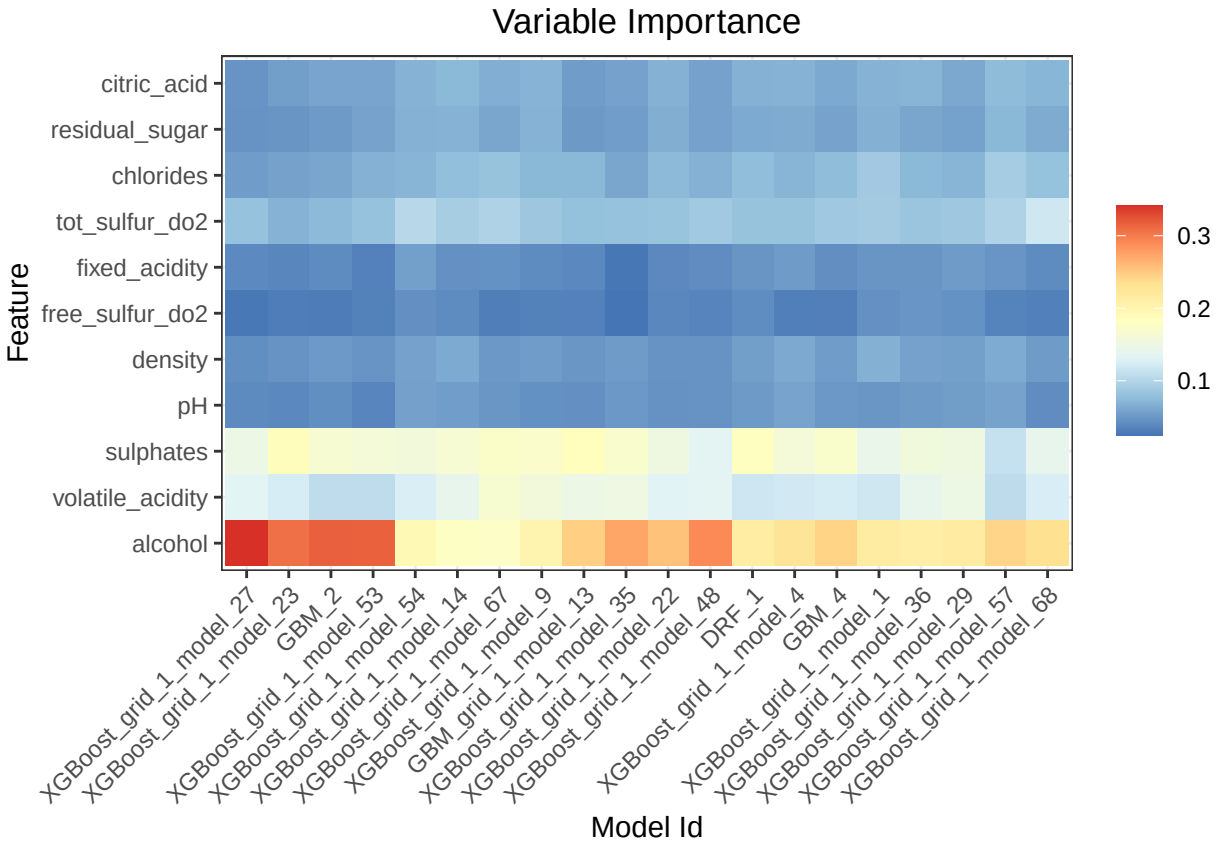
```
## | **10** |StackedEnsemble_AllModels_2_AutoML_4_20220328_133556 | 0.362639925909703 | 0.60219591987135
## | **11** |XGBoost_grid_1_AutoML_4_20220328_133556_model_67 | 0.363688133275926 | 0.603065612745352 |
## | **12** |XGBoost_grid_1_AutoML_4_20220328_133556_model_36 | 0.366022135574077 | 0.604997632701217 |
## | **13** |XGBoost_grid_1_AutoML_4_20220328_133556_model_27 | 0.367012264570853 | 0.60581537168584 | 
## | **14** |XGBoost_grid_1_AutoML_4_20220328_133556_model_9 | 0.367617424479608 | 0.606314624992345 | 
## | **15** |XGBoost_grid_1_AutoML_4_20220328_133556_model_4 | 0.370844682749144 | 0.6089701821511 | 0.3
## | **16** |DRF_1_AutoML_4_20220328_133556 | 0.371111879392238 | 0.609189526660003 | 0.371111879392238
## | **17** |GBM_4_AutoML_4_20220328_133556 | 0.372002651617509 | 0.609920201024289 | 0.372002651617509
## | **18** |XGBoost_grid_1_AutoML_4_20220328_133556_model_54 | 0.373246707120016 | 0.610939200837543 |
## | **19** |XGBoost_grid_1_AutoML_4_20220328_133556_model_57 | 0.373336975083348 | 0.61101307275978 | 
## | **20** |XGBoost_grid_1_AutoML_4_20220328_133556_model_68 | 0.37340669661824 | 0.611070124141444 | 
##
##
## Residual Analysis
## ==================
##
## > Residual Analysis plots the fitted values vs residuals on a test dataset. Ideally, residuals should
```



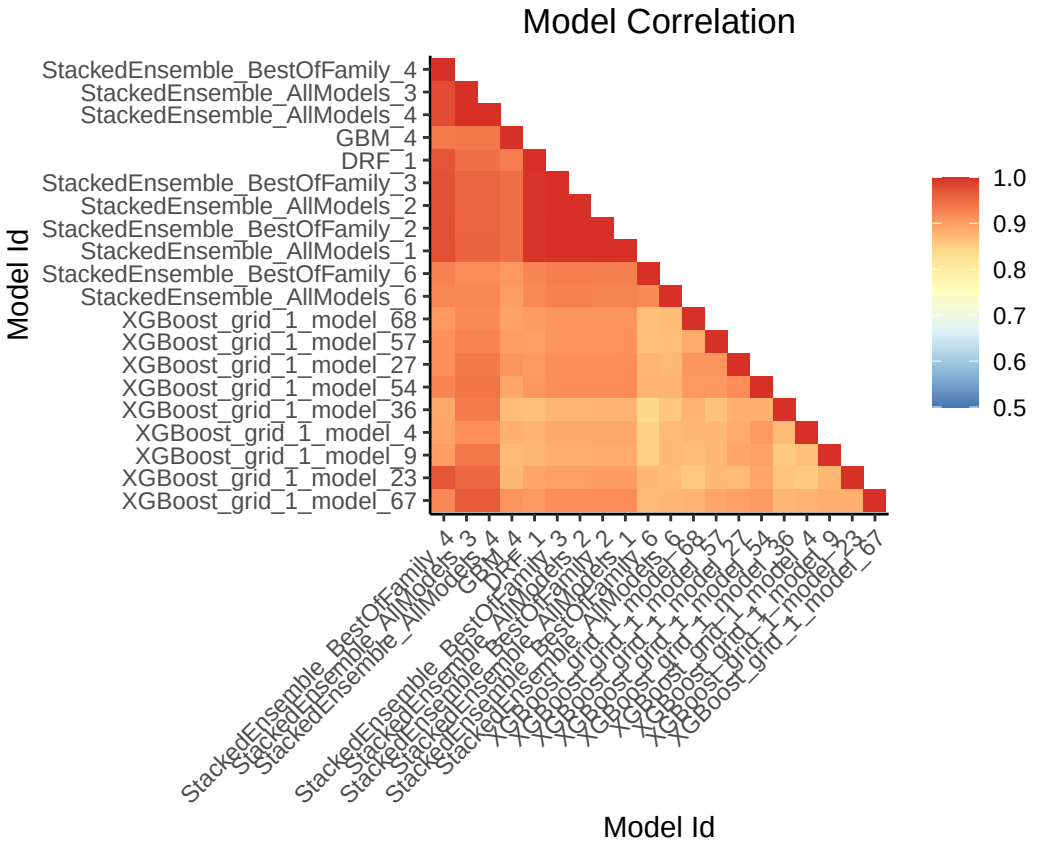

```
##
##
## Variable Importance
## ===================
##
## > The variable importance plot shows the relative importance of the most important variables in the m
```



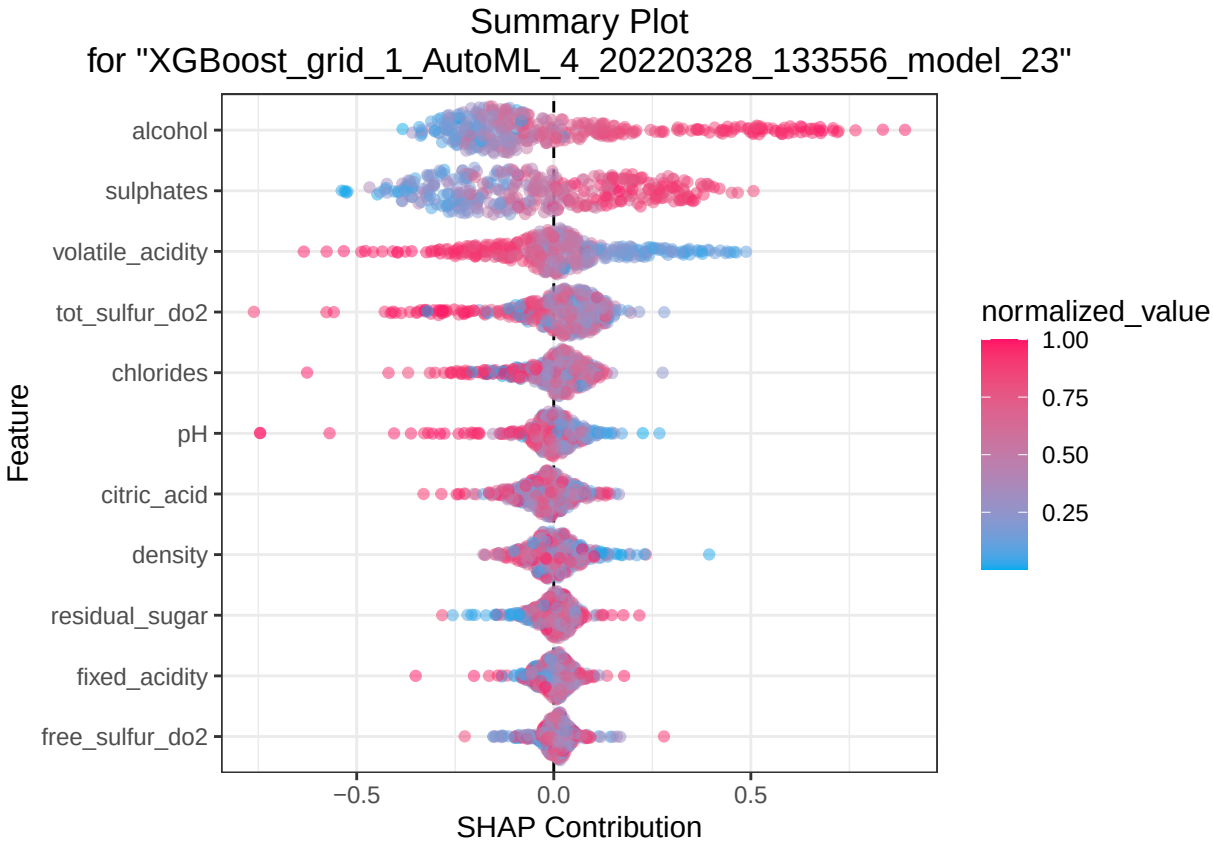

```
##
##
## Variable Importance Heatmap
## ==========================
##
## > Variable importance heatmap shows variable importance across multiple models. Some models in H2O r
```



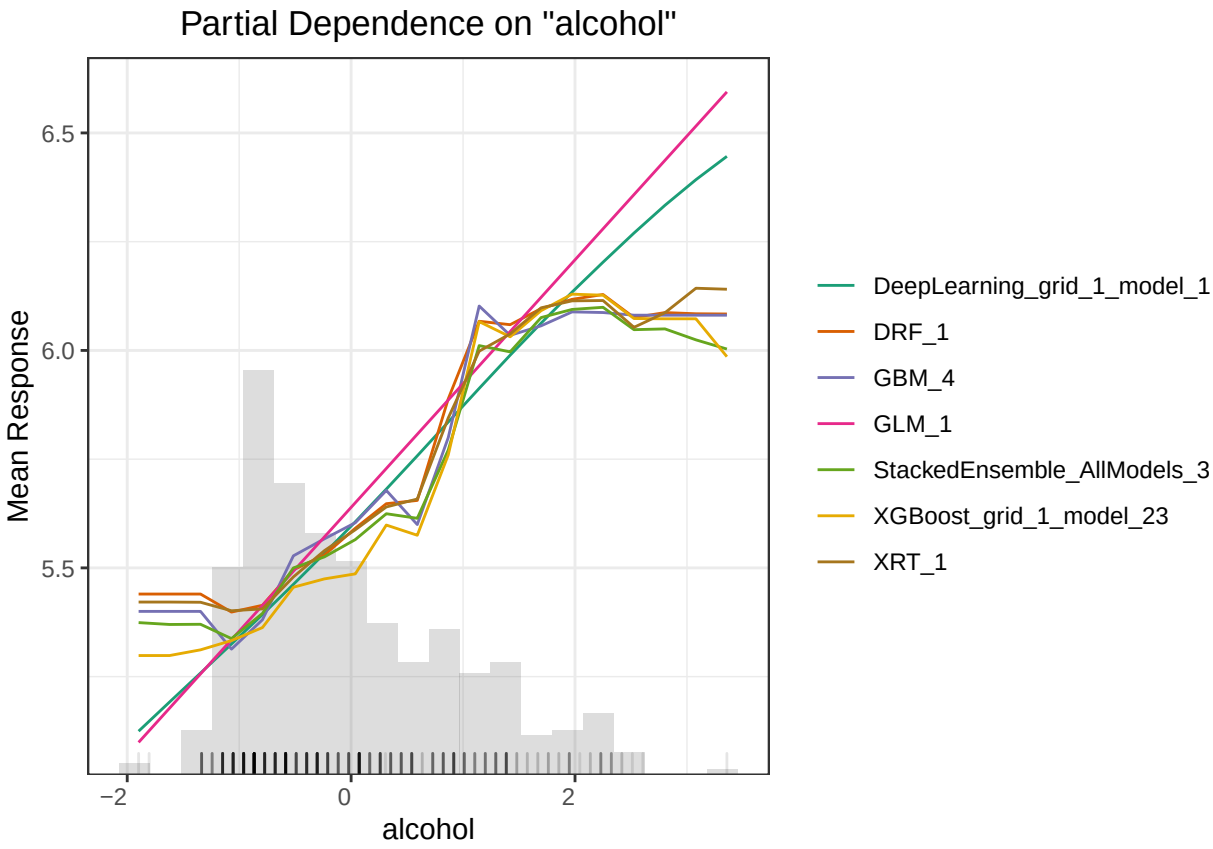

```
##
##
## Model Correlation
## =================
##
## > This plot shows the correlation between the predictions of the models. For classification, frequen
```

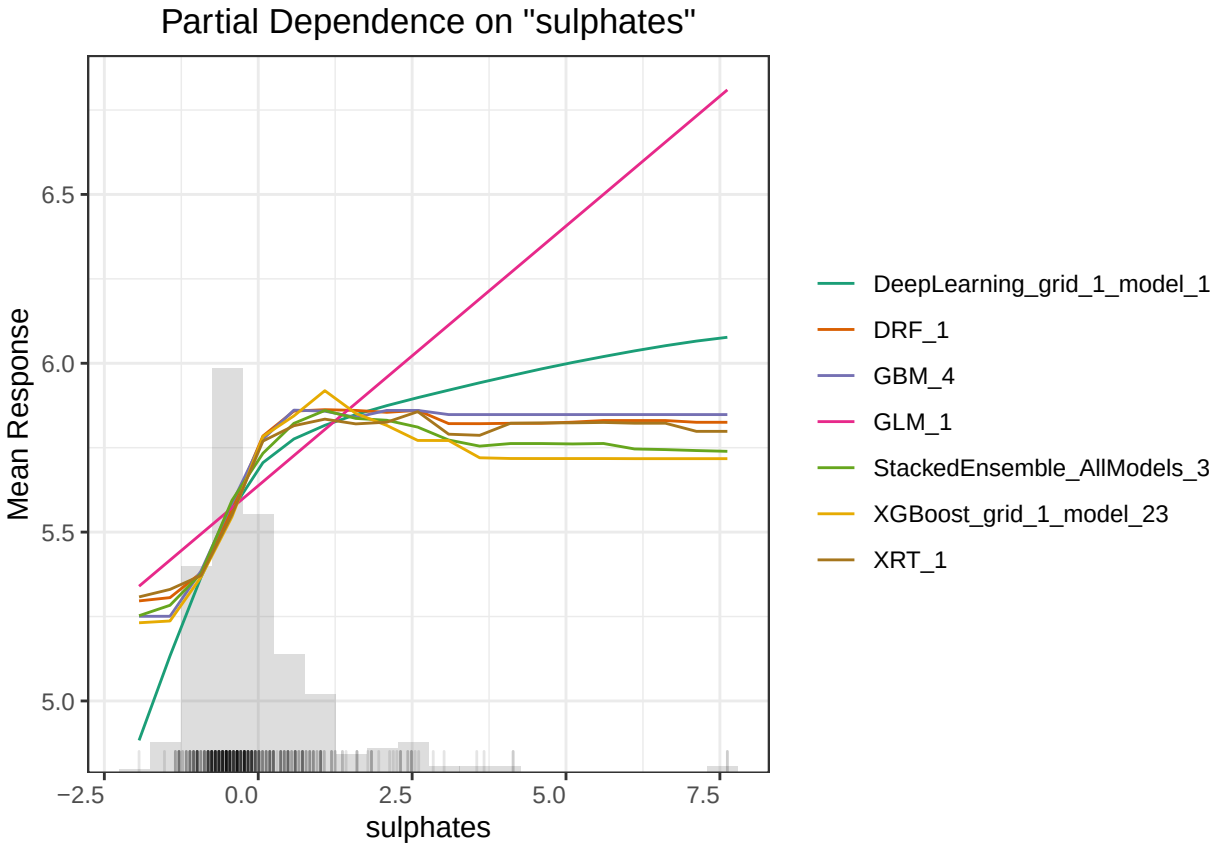


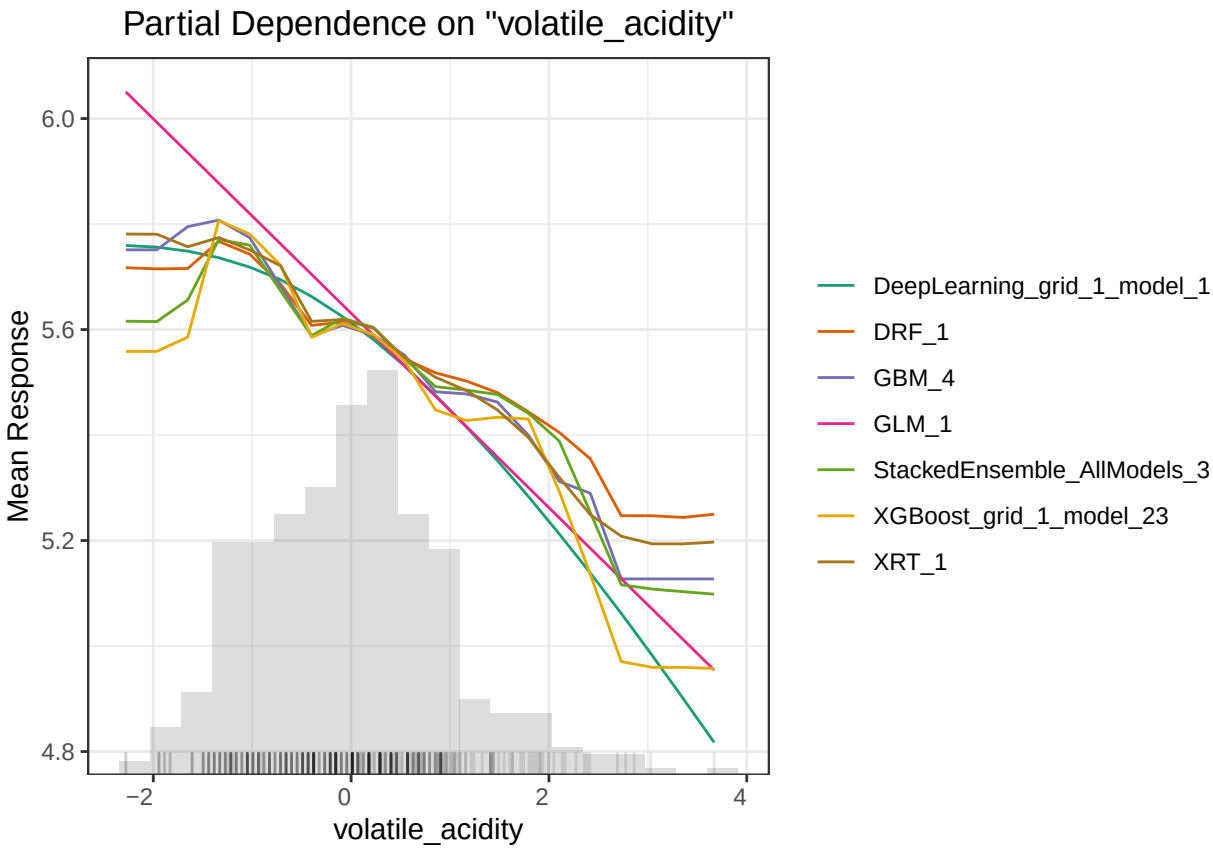

```
## Interpretable models: GLM_1_AutoML_4_20220328_133556
##
##
## SHAP Summary
## ============
##
## > SHAP summary plot shows the contribution of the features for each instance (row of data). The sum
```

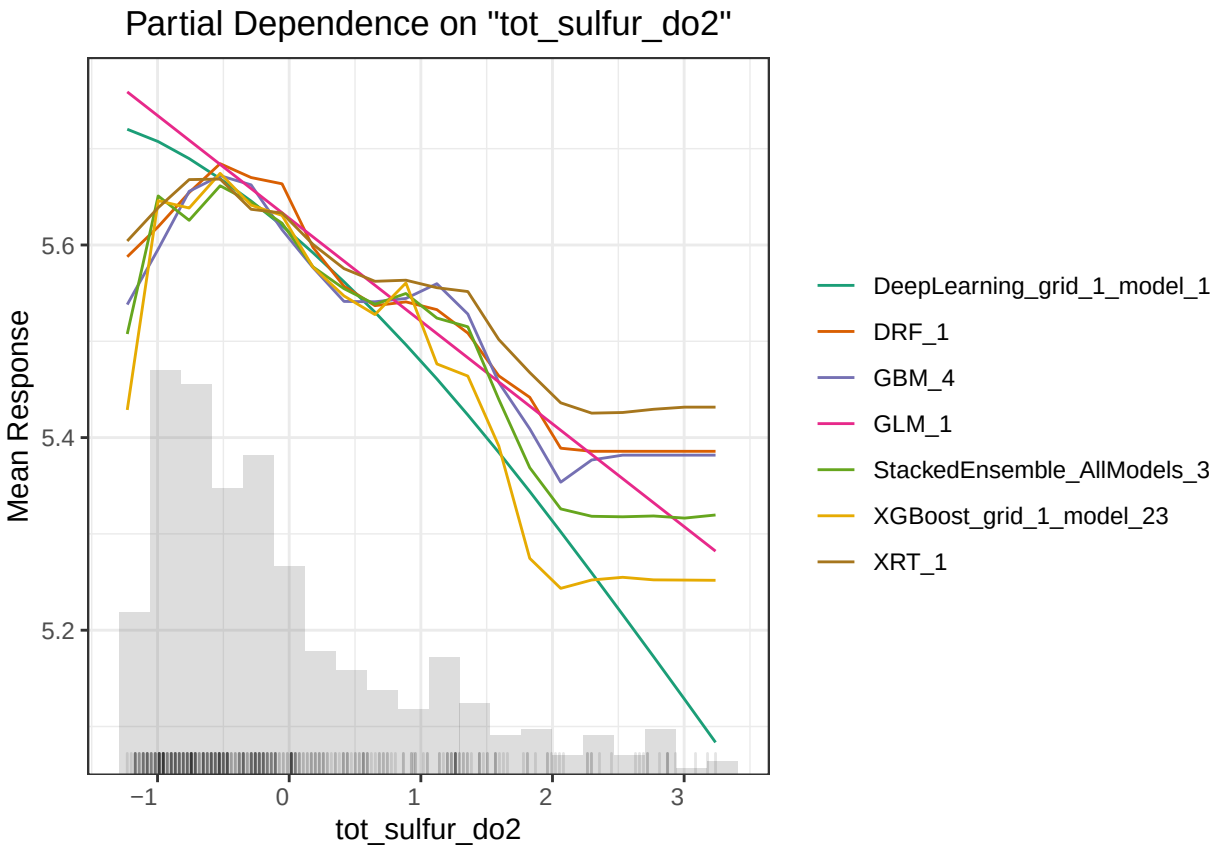


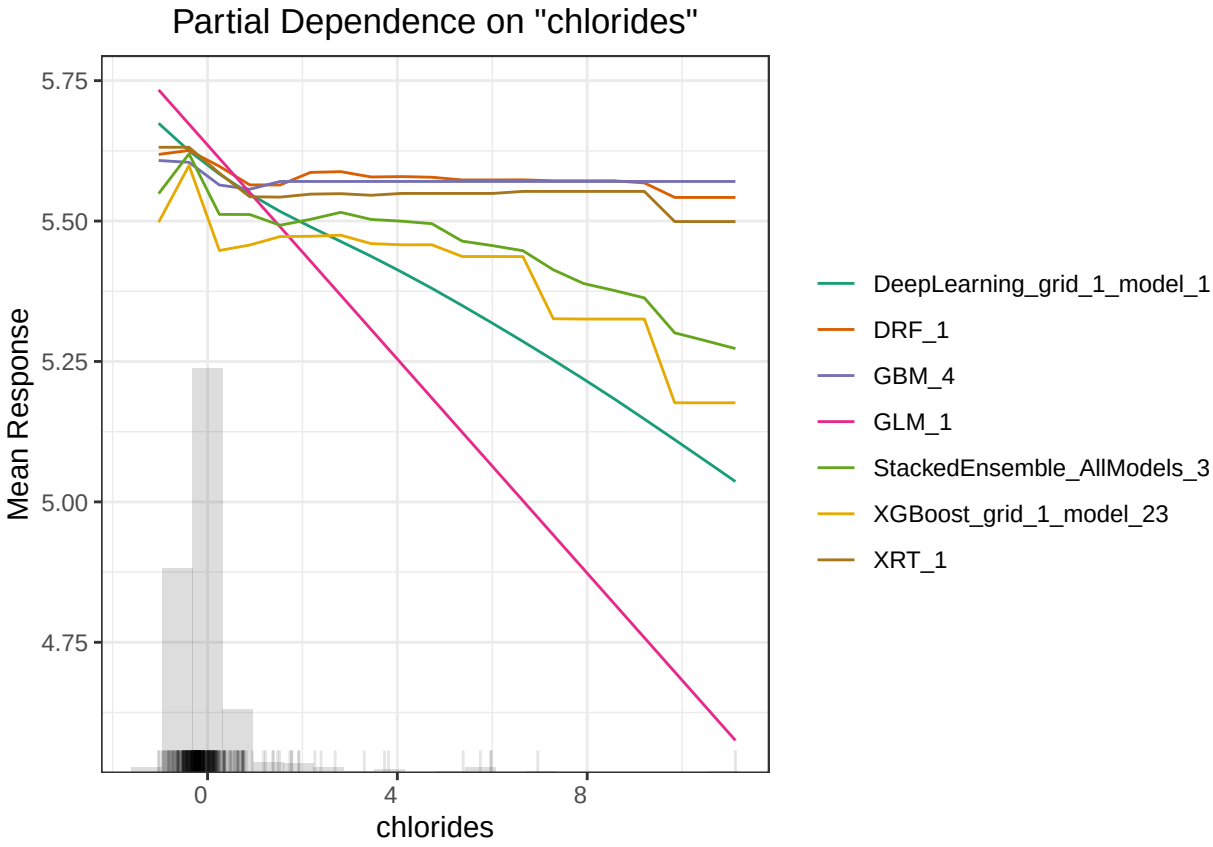

```
##
##
## Partial Dependence Plots
## ========================
##
## > Partial dependence plot (PDP) gives a graphical depiction of the marginal effect of a variable on
```


Partial Dependence on "alcohol"
Mean Response
6.5
6.0
5.5
−2
0
2
alcohol
DeepLearning_grid_1_model_1
DRF_1
GBM_4
GLM_1
StackedEnsemble_AllModels_3
XGBoost_grid_1_model_23
XRT_1

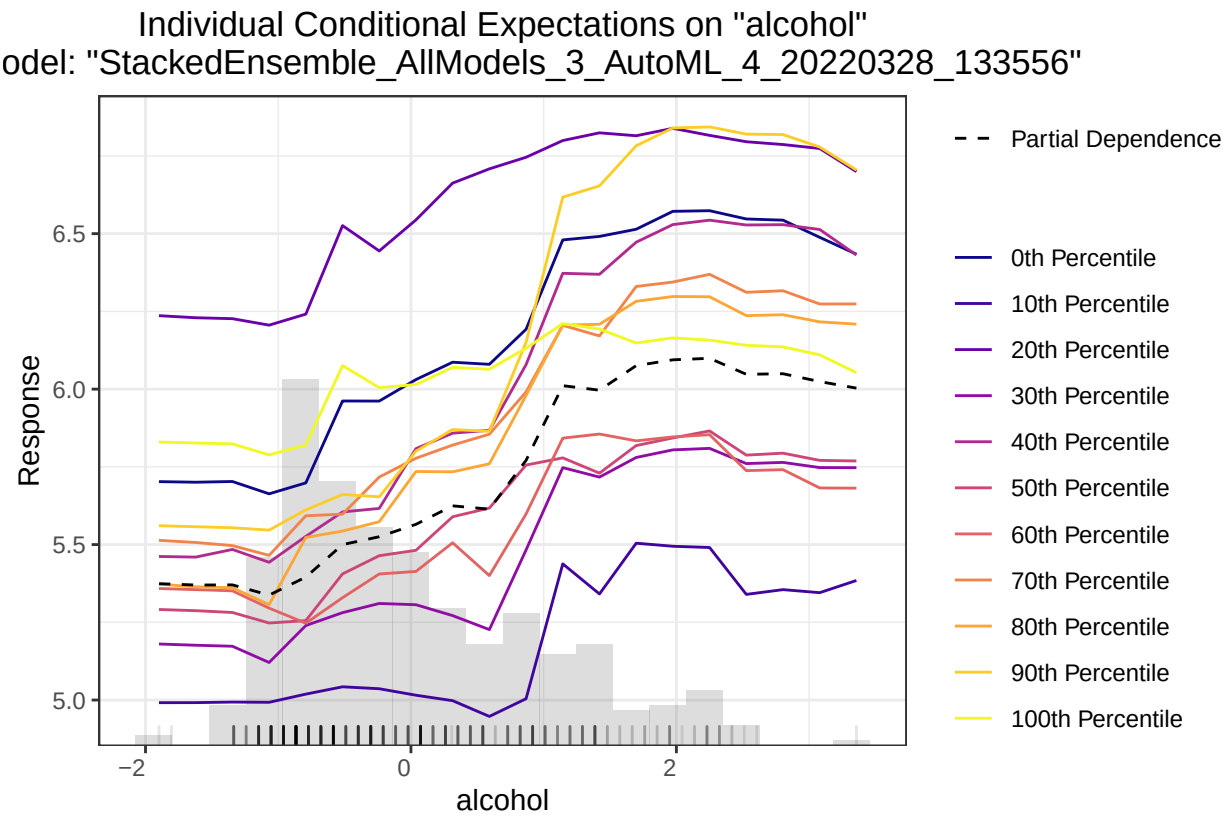

Partial Dependence on "sulphates"
Mean Response
6.5
6.0
5.5
5.0
−2.5
0.0
2.5
5.0
7.5
sulphates
DeepLearning_grid_1_model_1
DRF_1
GBM_4
GLM_1
StackedEnsemble_AllModels_3
XGBoost_grid_1_model_23
XRT_1

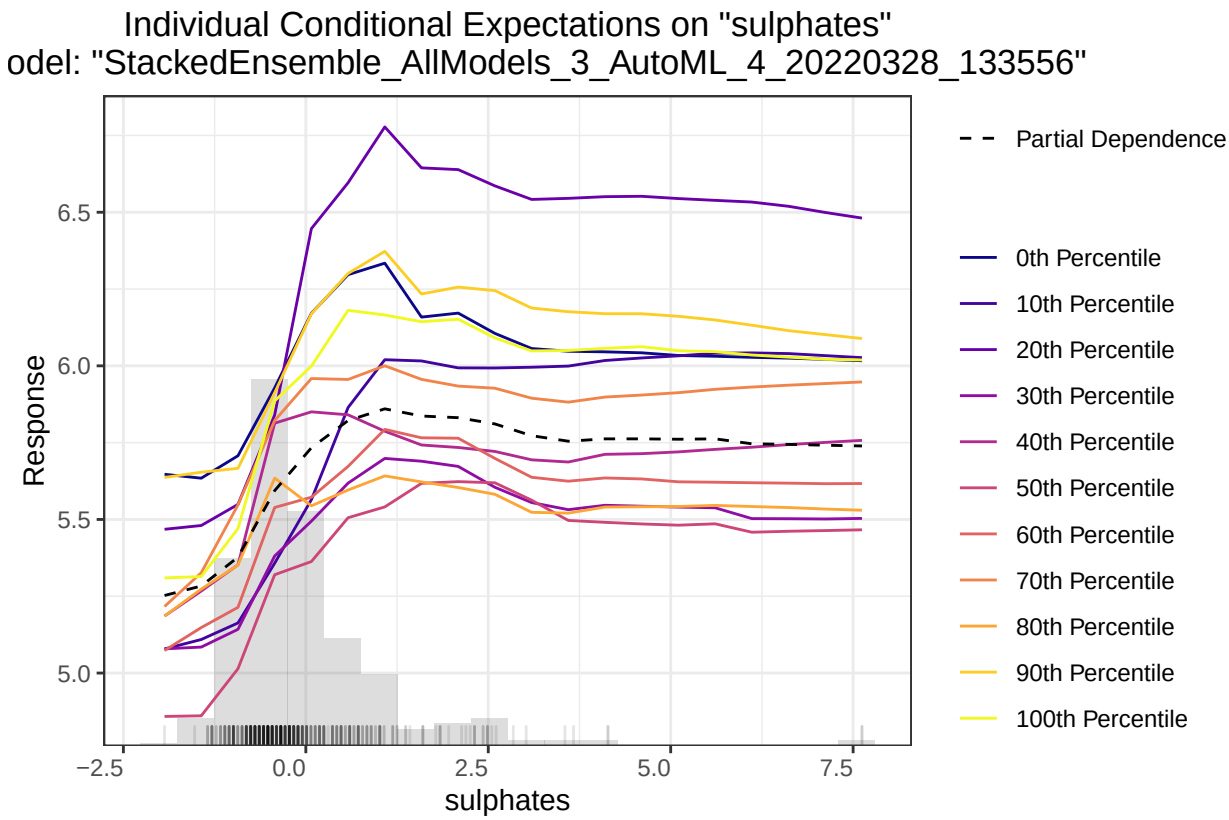

Partial Dependence on "volatile_acidity"
Mean Response
6.0
5.6
5.2
4.8
−2
0
2
4
volatile_acidity
DeepLearning_grid_1_model_1
DRF_1
GBM_4
GLM_1
StackedEnsemble_AllModels_3
XGBoost_grid_1_model_23
XRT_1

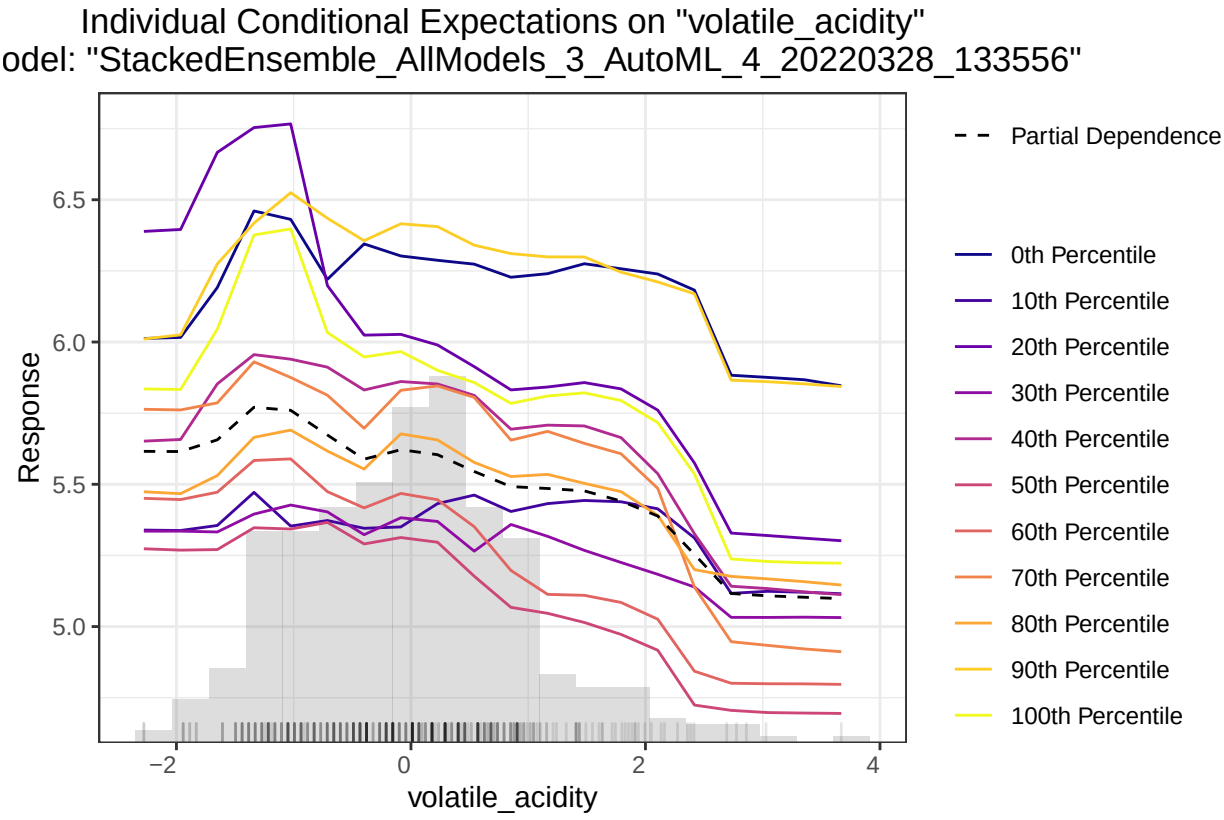

Partial Dependence on "tot_sulfur_do2"
Mean Response
5.6
5.4
5.2
−1
0
1
2
3
tot_sulfur_do2
DeepLearning_grid_1_model_1
DRF_1
GBM_4
GLM_1
StackedEnsemble_AllModels_3
XGBoost_grid_1_model_23
XRT_1

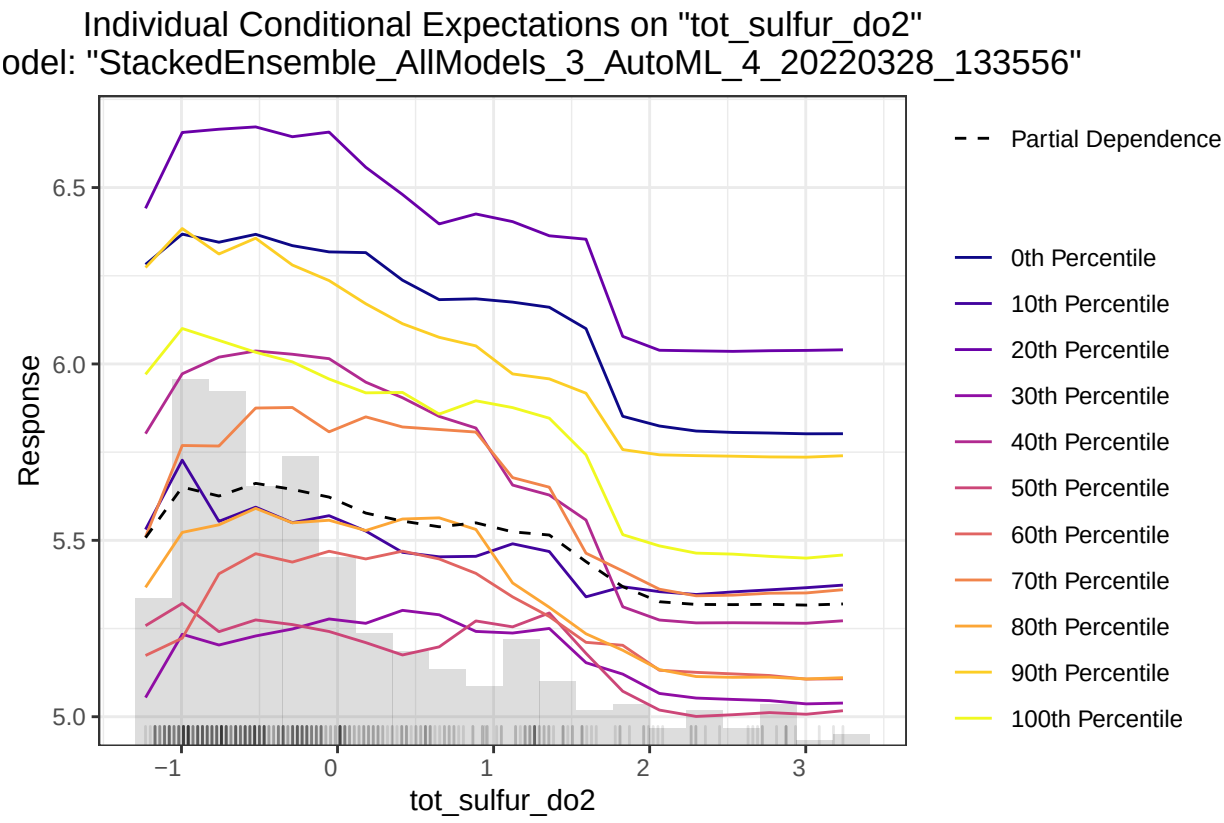


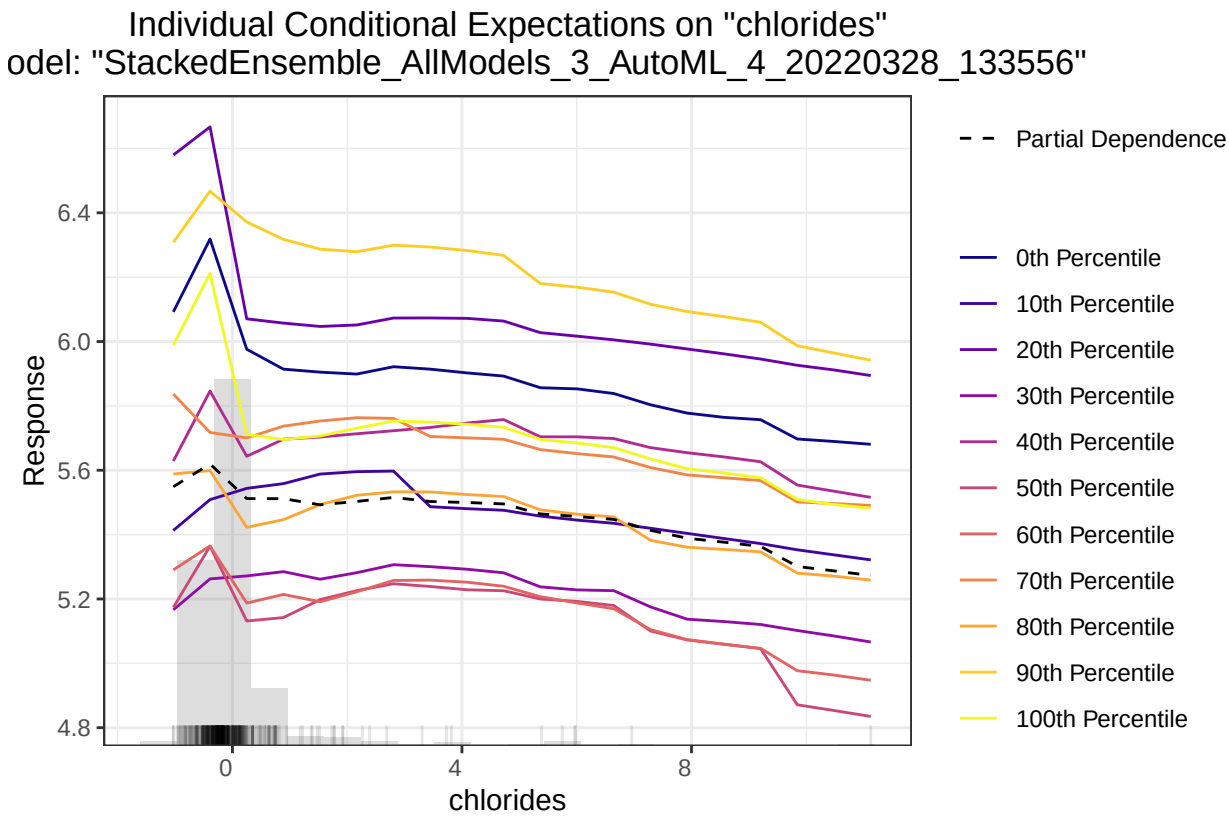

```
##
##
## Individual Conditional Expectations
## ==================================
##
## > An Individual Conditional Expectation (ICE) plot gives a graphical depiction of the marginal effec
```


Individual Conditional Expectations on "alcohol"
odel: "StackedEnsemble_AllModels_3_AutoML_4_20220328_133556"
Response
6.5
6.0
5.5
5.0
−2
0
2
alcohol
Partial Dependence
0th Percentile
10th Percentile
20th Percentile
30th Percentile
40th Percentile
50th Percentile
60th Percentile
70th Percentile
80th Percentile
90th Percentile
100th Percentile


Individual Conditional Expectations on "sulphates"
odel: "StackedEnsemble_AllModels_3_AutoML_4_20220328_133556"
Response
6.5
6.0
5.5
5.0
−2.5
0.0
2.5
5.0
7.5
sulphates
Partial Dependence
0th Percentile
10th Percentile
20th Percentile
30th Percentile
40th Percentile
50th Percentile
60th Percentile
70th Percentile
80th Percentile
90th Percentile
100th Percentile


Individual Conditional Expectations on "volatile_acidity"
odel: "StackedEnsemble_AllModels_3_AutoML_4_20220328_133556"
Response
6.5
6.0
5.5
5.0
−2
0
2
4
volatile_acidity
Partial Dependence
0th Percentile
10th Percentile
20th Percentile
30th Percentile
40th Percentile
50th Percentile
60th Percentile
70th Percentile
80th Percentile
90th Percentile
100th Percentile


Individual Conditional Expectations on "tot_sulfur_do2"
odel: "StackedEnsemble_AllModels_3_AutoML_4_20220328_133556"
Response
6.5
6.0
5.5
5.0
−1
0
1
2
3
tot_sulfur_do2
Partial Dependence
0th Percentile
10th Percentile
20th Percentile
30th Percentile
40th Percentile
50th Percentile
60th Percentile
70th Percentile
80th Percentile
90th Percentile
100th Percentile




```
library(lime)
model_ids <-as.data.frame(auto_ml@leaderboard$model_id)[,1]
best_model <- h2o.getModel(grep("StackedEnsemble_BestOfFamily", model_ids, value=TRUE)[1])

explainer <- lime(as.data.frame(training[, -12]), best_model, bin_continuous = F) #remove 'Attrition' c
explanation <- explain(as.data.frame(testing[, -12]), #cherry picked rows for explaining pusposes
                       explainer = explainer,
                       kernel_width = 1,
                       n_features = 5, #max features to explain each model
                       n_labels = 1)
```

```
## Warning in explain.data.frame(as.data.frame(testing[, -12]), explainer =
## explainer, : "labels" and "n_labels" arguments are ignored when explaining
## regression models
```

```
##   |                                                                      |
##   |                                                                      |
```

```
plot_features(explanation)
```

Model Evaluation

Conclusion

References

- P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.