Forestfire Regresyon Problemi

Final Ödevi

Deniz BALCI

Orman yangınlari veriseti(Regresyon problemi)

problem tanımı

Bu verisetindeki amacımız yanmış alanı tahmin etmektir.

Veri kaynağı: https://archive.ics.uci.edu/dataset/162/forest+fires

1. X - Montesinho park haritası içindeki x ekseni uzamsal koordinatı: 1'den 9'a kadar
2. Y - Montesinho park haritası içinde y ekseni uzamsal koordinatı: 2 ila 9
3. ay - yılın ayı: 'jan' ila 'dec'
4. gün - haftanın günü: 'mon' ile 'sun' arası
5. FFMC - FWI sisteminden FFMC endeksi: 18,7 ila 96,20
6. DMC - FWI sisteminden alınan DMC endeksi: 1,1 ila 291,3
7. DC - FWI sisteminden alınan DC endeksi: 7,9 ila 860,6
8. ISI - FWI sisteminden ISI endeksi: 0,0 ila 56,10
9. temp - Santigrat derece cinsinden sıcaklık: 2,2 ila 33,30
10. RH - % cinsinden bağıl nem: 15,0 ila 100
11. rüzgar - km/sa cinsinden rüzgar hızı: 0.40 ila 9.40
12. yağmur - mm/m2 cinsinden dış yağmur: 0.0 ila 6.4
13. alan - ormanın yanmış alanı (hektar olarak): 0.00 ila 1090.84 (bu çıktı değişkeni 0.0'a doğru çok çarpıktır, bu nedenle logaritma dönüşümü ile modellemek mantıklıdır).

Veri ön işleme

Paketlerin yüklenmesi

```
library(caret)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```
library(xgboost)
library(caret)
library(lightgbm)
```

```

Attaching package: 'lightgbm'

The following object is masked from 'package:xgboost':

    slice
```

```
library(rsample)
library(party)
```

```
Loading required package: grid

Loading required package: mvtnorm

Loading required package: modeltools

Loading required package: stats4

Loading required package: strucchange

Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich
```

```r
library(caTools)
library(randomForest)
```

```
randomForest 4.7-1.1
```

```
Type rfNews() to see new features/changes/bug fixes.
```

```
Attaching package: 'randomForest'
```

```
The following object is masked from 'package:ggplot2':
```

```
    margin
```

```r
library(corrplot)
```

```
corrplot 0.92 loaded
```

```r
library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following object is masked from 'package:randomForest':
```

```
    combine
```

```
The following object is masked from 'package:party':
```

```
    where
```

```
The following object is masked from 'package:lightgbm':
```

```
    slice
```

```
The following object is masked from 'package:xgboost':

    slice

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
df <- read.csv('/home/deniz/Masaüstü/yükseklisans/ayz/xaifinal/forestfires.csv', sep=',')

# Get unique values for each column
unique_values <- sapply(df, function(y) sum(length(unique(y))))
head(unique_values)
```

```
  X     Y month   day  FFMC   DMC
  9     7    12     7   106   215
```

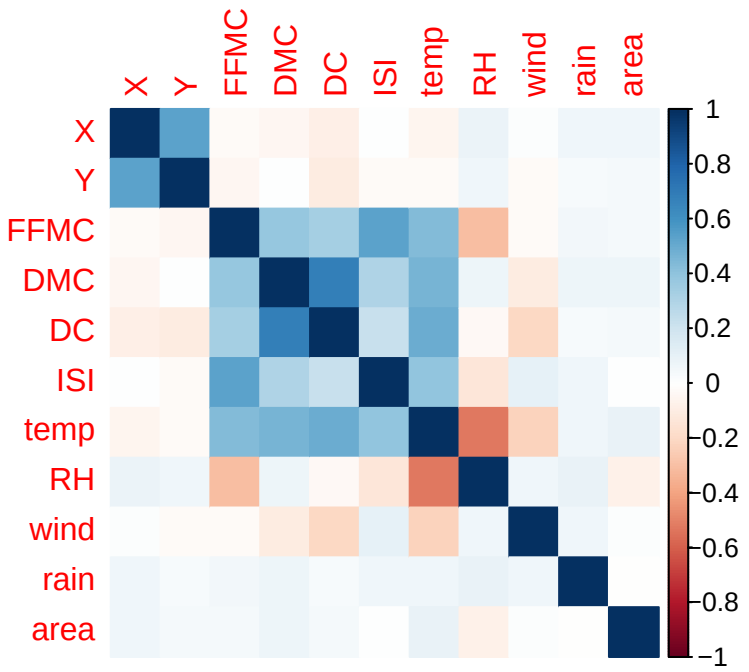
```r
# Display unique values
na_count <-sapply(df, function(y) sum(length(which(is.na(y)))))
head(na_count)
```

```
  X     Y month   day  FFMC   DMC
  0     0     0     0     0     0
```

```r
categorical_cols <- df %>% select_if(is.character)
head(categorical_cols)
```

```
  month day
1   mar fri
2   oct tue
3   oct sat
4   mar fri
5   mar sun
6   aug sun
```

```r
# Select numerical columns
numerical_cols <- df %>% select_if(is.numeric)


corr_matrix <- cor(numerical_cols )

corrplot::corrplot(corr_matrix, method = "color")
```



veri parçalama

```r
# Split into training (70%) and testing set (30%)
parts <- createDataPartition(df$area, p = 0.7, list = FALSE)


train <- df[parts, ]
test <- df[-parts, ]

# Define predictor and response variables in the training set
train_x <- data.matrix(train[, -which(names(train) == 'area')])
train_y <- train[['area']]
```

```r
# Define predictor and response variables in the testing set
test_x <- data.matrix(test[, -which(names(test) == 'area')])
test_y <- test[['area']]
```

BLACKBOX MODELLER

Aşağıdaki açıklamak için 3 farklı model kullanacağız bu bölümde modeller oluşturulacaktır.
XBOOST

```r
# Define final training and testing sets
xgb_train <- xgb.DMatrix(data = train_x, label = train_y)
xgb_test <- xgb.DMatrix(data = test_x, label = test_y)

# Define watchlist
watchlist <- list(train = xgb_train, test = xgb_test)

# Fit XGBoost model and display training and testing data at each round
model <- xgb.train(data = xgb_train, max.depth = 3, watchlist = watchlist, nrounds = 70, o
```

```
[1] train-rmse:67.682428    test-rmse:21.908867
[2] train-rmse:61.194436    test-rmse:22.800268
[3] train-rmse:56.160732    test-rmse:23.131703
[4] train-rmse:52.661168    test-rmse:26.124551
[5] train-rmse:48.722280    test-rmse:27.741769
[6] train-rmse:45.593632    test-rmse:28.514046
[7] train-rmse:42.921328    test-rmse:32.430345
[8] train-rmse:40.584419    test-rmse:32.682258
[9] train-rmse:38.177613    test-rmse:32.985821
[10]    train-rmse:36.041509    test-rmse:35.239242
[11]    train-rmse:34.411412    test-rmse:34.736753
[12]    train-rmse:32.950517    test-rmse:36.825519
[13]    train-rmse:31.398830    test-rmse:37.093981
[14]    train-rmse:30.374555    test-rmse:37.280762
[15]    train-rmse:28.732403    test-rmse:37.108459
[16]    train-rmse:27.738469    test-rmse:37.597954
[17]    train-rmse:27.132646    test-rmse:37.474830
[18]    train-rmse:26.366780    test-rmse:37.707886
[19]    train-rmse:25.880695    test-rmse:37.360117
[20]    train-rmse:25.203151    test-rmse:37.740245
[21]    train-rmse:23.969276    test-rmse:37.559818
```

```
[22]    train-rmse:22.736642    test-rmse:37.509607
[23]    train-rmse:22.359121    test-rmse:37.244913
[24]    train-rmse:22.118872    test-rmse:37.500224
[25]    train-rmse:21.699242    test-rmse:37.599427
[26]    train-rmse:21.411891    test-rmse:38.137624
[27]    train-rmse:20.543480    test-rmse:39.107216
[28]    train-rmse:20.006608    test-rmse:39.278162
[29]    train-rmse:19.478627    test-rmse:39.158466
[30]    train-rmse:18.780187    test-rmse:39.171789
[31]    train-rmse:18.529866    test-rmse:39.491736
[32]    train-rmse:18.049611    test-rmse:39.805859
[33]    train-rmse:17.834435    test-rmse:40.716070
[34]    train-rmse:17.471120    test-rmse:40.879931
[35]    train-rmse:17.239826    test-rmse:41.595157
[36]    train-rmse:17.072987    test-rmse:41.684680
[37]    train-rmse:16.645170    test-rmse:41.617660
[38]    train-rmse:16.532223    test-rmse:41.620508
[39]    train-rmse:16.417010    test-rmse:41.489057
[40]    train-rmse:16.117951    test-rmse:41.434575
[41]    train-rmse:15.848955    test-rmse:41.342245
[42]    train-rmse:15.771998    test-rmse:41.344755
[43]    train-rmse:15.568584    test-rmse:41.302286
[44]    train-rmse:15.382945    test-rmse:41.425517
[45]    train-rmse:15.287118    test-rmse:41.463351
[46]    train-rmse:15.142174    test-rmse:41.457358
[47]    train-rmse:14.969452    test-rmse:41.417649
[48]    train-rmse:14.894735    test-rmse:41.524491
[49]    train-rmse:14.320939    test-rmse:41.806531
[50]    train-rmse:14.148476    test-rmse:41.681424
[51]    train-rmse:13.712226    test-rmse:42.018521
[52]    train-rmse:13.620914    test-rmse:41.881199
[53]    train-rmse:13.524812    test-rmse:41.938985
[54]    train-rmse:13.025379    test-rmse:42.018115
[55]    train-rmse:12.872613    test-rmse:42.043178
[56]    train-rmse:12.805716    test-rmse:42.031781
[57]    train-rmse:12.626518    test-rmse:42.095065
[58]    train-rmse:12.473717    test-rmse:42.294344
[59]    train-rmse:12.417053    test-rmse:42.304182
[60]    train-rmse:12.383391    test-rmse:42.270117
[61]    train-rmse:12.052947    test-rmse:42.396589
[62]    train-rmse:11.943671    test-rmse:42.175739
[63]    train-rmse:11.683088    test-rmse:42.309264
[64]    train-rmse:11.608244    test-rmse:42.385798
```

```
[65]    train-rmse:11.323745    test-rmse:42.394236
[66]    train-rmse:11.270778    test-rmse:42.436178
[67]    train-rmse:11.060163    test-rmse:42.565949
[68]    train-rmse:10.847006    test-rmse:42.596884
[69]    train-rmse:10.763434    test-rmse:42.735575
[70]    train-rmse:10.742235    test-rmse:42.734972
```

```r
# Make predictions on the testing set
pred_y <- predict(model, xgb_test)

# Calculate MSE, MAE, and RMSE
mse <- mean((test_y - pred_y)^2)
mae <- caret::MAE(test_y, pred_y)
rmse <- caret::RMSE(test_y, pred_y)

print(paste("MSE:", mse))
```

```
[1] "MSE: 1826.27784879151"
```

```r
print(paste("MAE:", mae))
```

```
[1] "MAE: 19.5748211185254"
```

```r
print(paste("RMSE:", rmse))
```

```
[1] "RMSE: 42.7349721983239"
```

LİGHTGBM

```r
dtrain = lgb.Dataset(train_x, label = train_y)
dtest = lgb.Dataset.create.valid(dtrain, test_x, label = test_y)

# define parameters
params = list(
  objective = "regression"
  , metric = "l2"
```

```
  , min_data = 1L
  , learning_rate = .3
)

# validataion data
valids = list(test = dtest)


# train model
model1 = lgb.train(
  params = params
  , data = dtrain
  , nrounds = 5L
  , valids = valids
)
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.0001
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 509
[LightGBM] [Info] Number of data points in the train set: 364, number of used features: 12
[LightGBM] [Info] Start training from score 15.016676
[1]:  test's l2:2459.98
[2]:  test's l2:5859.19
[3]:  test's l2:9167.5
[4]:  test's l2:11918.4
[5]:  test's l2:14135.2
```

```
lgb.get.eval.result(model1, "test", "l2")
```

```
[1]  2459.985  5859.187  9167.498 11918.446 14135.232
```

```
# prediction
pred_y1 = predict(model1, test_x)


# accuracy check
mse1 = mean((test_y - pred_y1)^2)
mae1 = caret::MAE(test_y, pred_y1)
rmse1 = caret::RMSE(test_y, pred_y1)
```

```r
cat("MSE: ", mse1, "\nMAE: ", mae1, "\nRMSE: ", rmse1)
```

```
MSE:  2459.985
MAE:  20.83134
RMSE:  49.59823
```

GBM MODEL

```r
library(gbm)
```

```
Loaded gbm 2.1.8.1
```

```r
train$month <- as.factor(train$month)
train$day <- as.factor(train$day)

# Assuming you have a data frame named 'train' with the response variable 'Diabetes_012'
gbmmodel <- gbm(area ~ ., data = train, distribution = "gaussian", n.trees = 20,
                shrinkage = 0.01, interaction.depth = 4)

# Assuming your 'test_x' is a data frame
# Convert it to a data frame if it's a matrix
test_x_df <- as.data.frame(test_x)

# Predict using the gbm model
pred_y2 <- predict(gbmmodel, newdata = test_x_df, n.trees = 20, type = "response")

# Assuming 'test_y' is the true response variable
# accuracy check
mse2 <- mean((test_y - pred_y2)^2)
mae2 <- caret::MAE(test_y, pred_y2)
rmse2 <- caret::RMSE(test_y, pred_y2)

cat("MSE: ", mse2, "\nMAE: ", mae2, "\nRMSE: ", rmse2)
```

```
MSE:  476.8477
MAE:  14.9188
RMSE:  21.83684
```

AÇIKLAYICI YAPAY ZEKA BÖLÜMÜ

İlk olarak local açıklayıcılar ile başlayacağız.Eğitim verisindeki 1.veri ile çalışmaya başlayacağız
LOCAL AÇIKLAYICILAR

XGBOOST MODELİ

BREAKDOWN

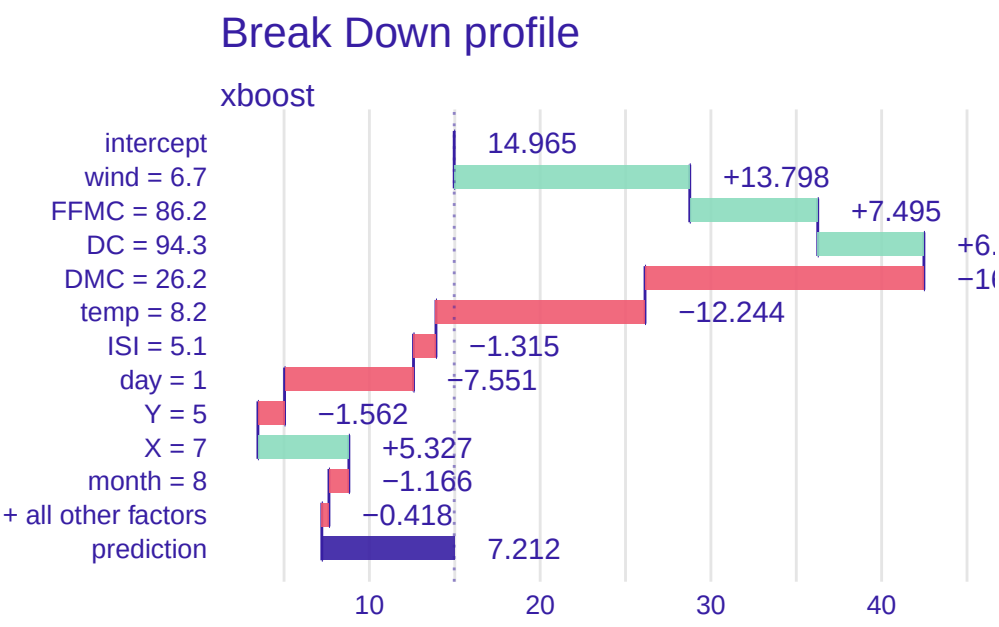
```
library(DALEX)
```

```
Welcome to DALEX (version: 2.4.3).
Find examples and detailed introduction at: http://ema.drwhy.ai/


Attaching package: 'DALEX'

The following object is masked from 'package:dplyr':

    explain
```

```
explain_xboost_breakdown <- DALEX::explain(model = model,
                                           data = train_x,
                                           y = train_y,
                                           label = "xboost")
```

```
Preparation of a new explainer is initiated
  -> model label       :  xboost
  -> data              :  364  rows  12  cols
  -> target variable   :  364  values
  -> predict function  :  yhat.default will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package Model of class: xgb.Booster package unrecognized , ver. Unl
  -> predicted values  :  numerical, min =  -11.86818 , mean =  14.96462 , max =  1034.895
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -45.70413 , mean =  0.05205815 , max =  66.76136
  A new explainer has been created!
```

```r
# Assuming train_x[[1]] is a numeric vector or matrix
new_observation <- as.data.frame(train_x[1, , drop = FALSE])

# Extract feature names from the XGBoost model
model_feature_names <- colnames(model$feature_names)

# Check and set correct feature names in the new observation
if (!identical(colnames(new_observation), model_feature_names)) {
  colnames(new_observation) <- model_feature_names
}

# Convert the new observation to a matrix
new_observation_matrix <- as.matrix(new_observation)

# Predict using the XGBoost model
prediction <- predict(model, newdata = new_observation_matrix)

# Calculate breakdown values
bd_xboost <- predict_parts(explain_xboost_breakdown, new_observation = new_observation_mat

# Plot breakdown values
plot(bd_xboost)
```



xboost yönteminde ve breakdown methodunda en çok katkıyı DC,DMC değişkenleri yap-

mıştır.

SHAPLEY DEĞERLERİ

```
explain_xboost_shap <- DALEX::explain(model = model,
                                      data = train_x,
                                      y = train_y,
                                      label = "xboost shapley değerleri")
```

```
Preparation of a new explainer is initiated
  -> model label       :  xboost shapley değerleri
  -> data              :  364  rows  12  cols
  -> target variable   :  364  values
  -> predict function  :  yhat.default will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package Model of class: xgb.Booster package unrecognized , ver. Unl
  -> predicted values  :  numerical, min =  -11.86818 , mean =  14.96462 , max =  1034.895
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -45.70413 , mean =  0.05205815 , max =  66.76136
  A new explainer has been created!
```

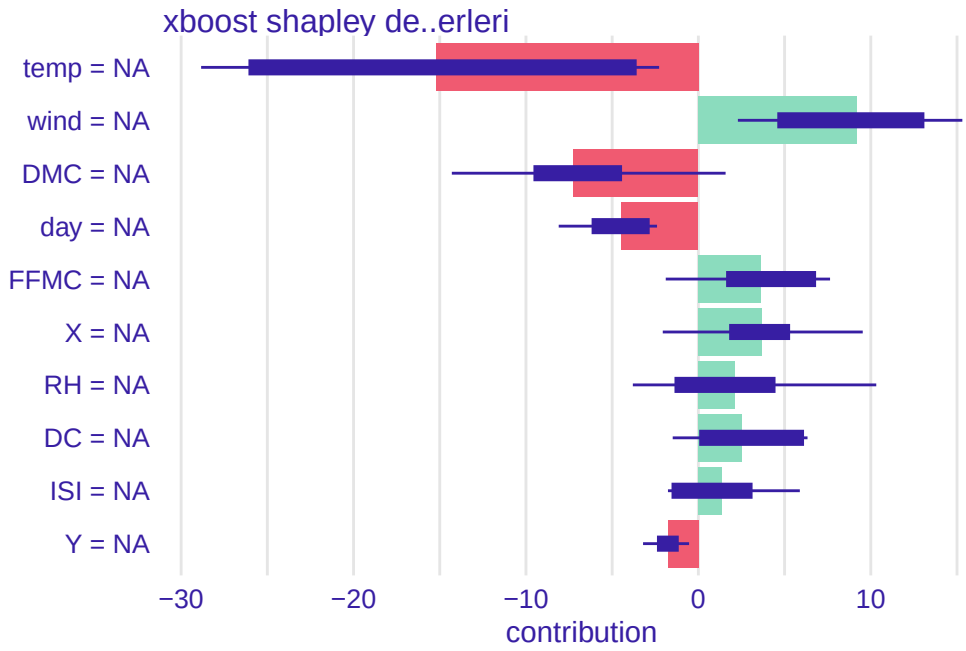
```
prediction_xboost <- predict(model, newdata = new_observation_matrix)

sh_xboost <- predict_parts(explain_xboost_shap, new_observation = new_observation_matrix,

plot(sh_xboost)
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'xboost shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>
```



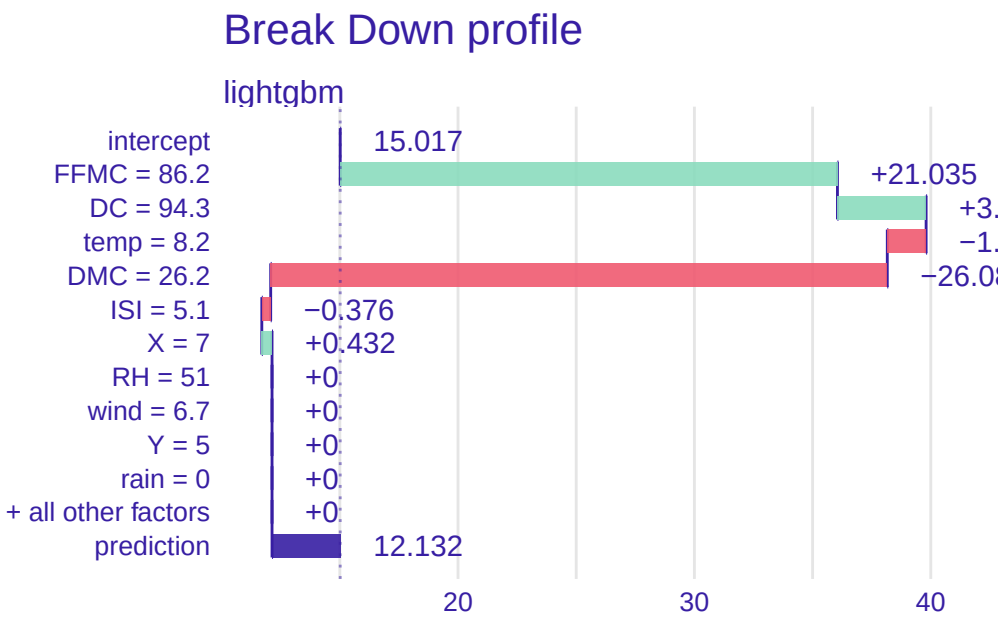
xboost yönteminde ve shapley methodunda en çok katkıyı DC,DMC değişkenleri yapmıştır.
LİGHTGBM MODELİ

BREAKDOWN

```
library(DALEX)
explain_lightbm_breakdown <- DALEX::explain(model = model1,
                                            data = train_x,
                                            y = train_y,
                                            label = "lightgbm")
```

```
Preparation of a new explainer is initiated
  -> model label       :  lightgbm
  -> data              :  364  rows  12  cols
  -> target variable   :  364  values
  -> predict function  :  yhat.default will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package Model of class: lgb.Booster package unrecognized , ver. Unl
  -> predicted values  :  numerical, min =  10.51167 , mean =  15.01668 , max =  337.7636
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -16.84876 , mean =  3.386657e-08 , max =  753.076
  A new explainer has been created!
```

```
# Calculate breakdown values
bd_lightbm <- predict_parts(explain_lightbm_breakdown, new_observation = new_observation_m

# Plot breakdown values
plot(bd_lightbm )
```




```
######################## breakdown ##############################
```

Lightgbm yönteminde ve breakdown methodunda en çok katkıyı Y,TEMP değişkenleri yapmıştır. #### SHAPLEY DEĞERLERİ

```
######################## shap ##############################

explain_lightgbm_shap <- DALEX::explain(model = model1,
                                        data = train_x,
                                        y = train_y,
                                        label = "lightgbm shapley değerleri")
```

```
Preparation of a new explainer is initiated
  -> model label       :  lightgbm shapley değerleri
  -> data              :  364  rows  12  cols
  -> target variable   :  364  values
```

```
  -> predict function  :  yhat.default will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package Model of class: lgb.Booster package unrecognized , ver. Unl
  -> predicted values  :  numerical, min =  10.51167 , mean =  15.01668 , max =  337.7636
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -16.84876 , mean =  3.386657e-08 , max =  753.076
  A new explainer has been created!
```

```r
sh_lightgbm <- predict_parts(explain_lightgbm_shap, new_observation = new_observation_matr

plot(sh_lightgbm )
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>
```

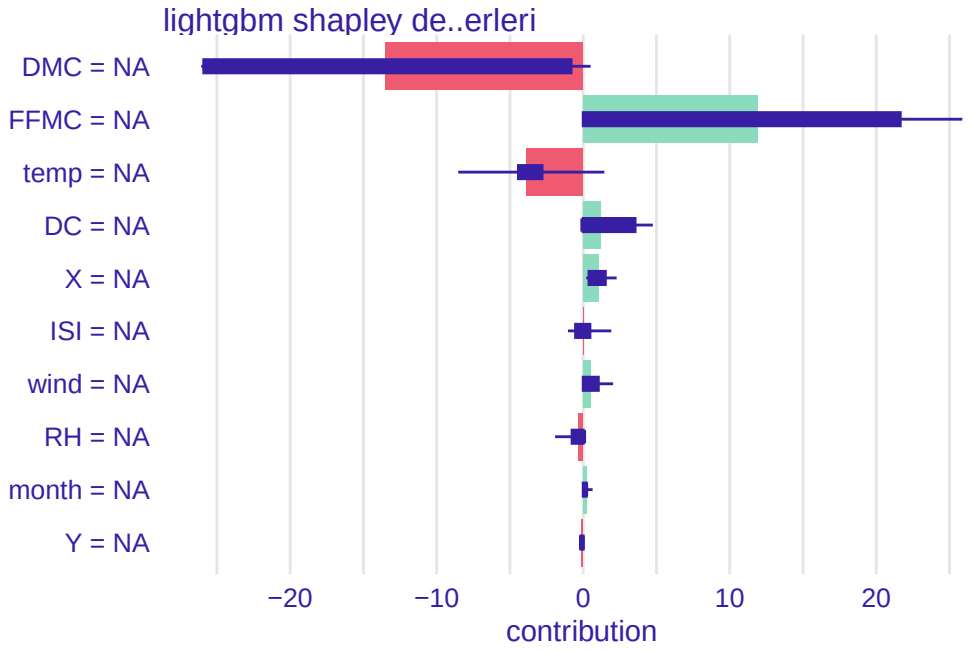
```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <c4>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'lightgbm shapley değerleri' in 'mbcsToSbcs': dot
substituted for <9f>
```



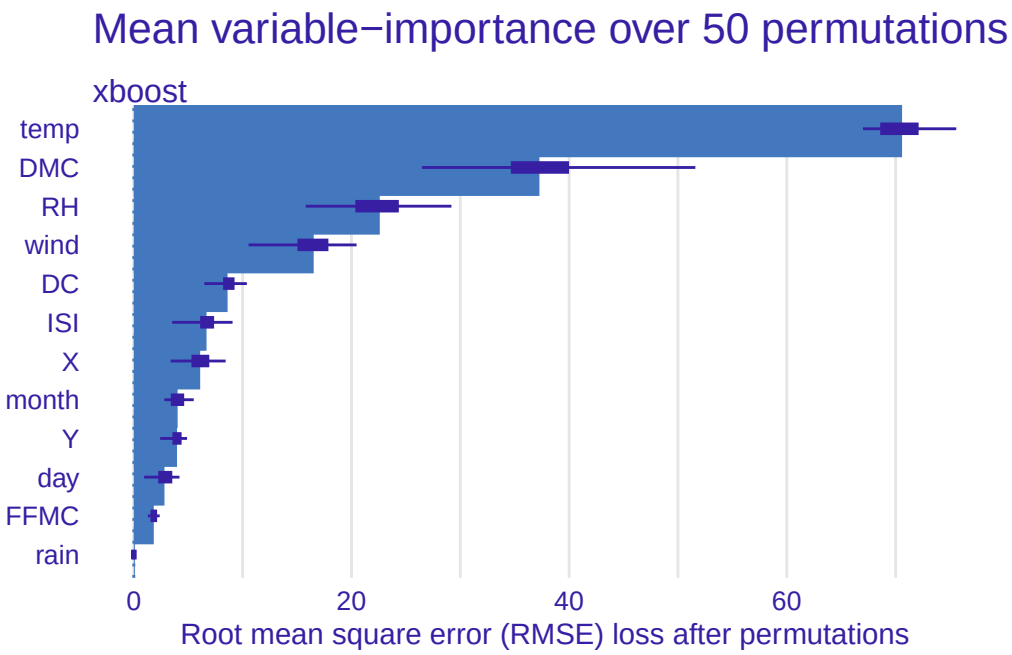

```
######################## shap ##############################
```

Lightgbm yönteminde ve shapley methodunda en çok katkıyı Y,TEMP değişkenleri yapmıştır.

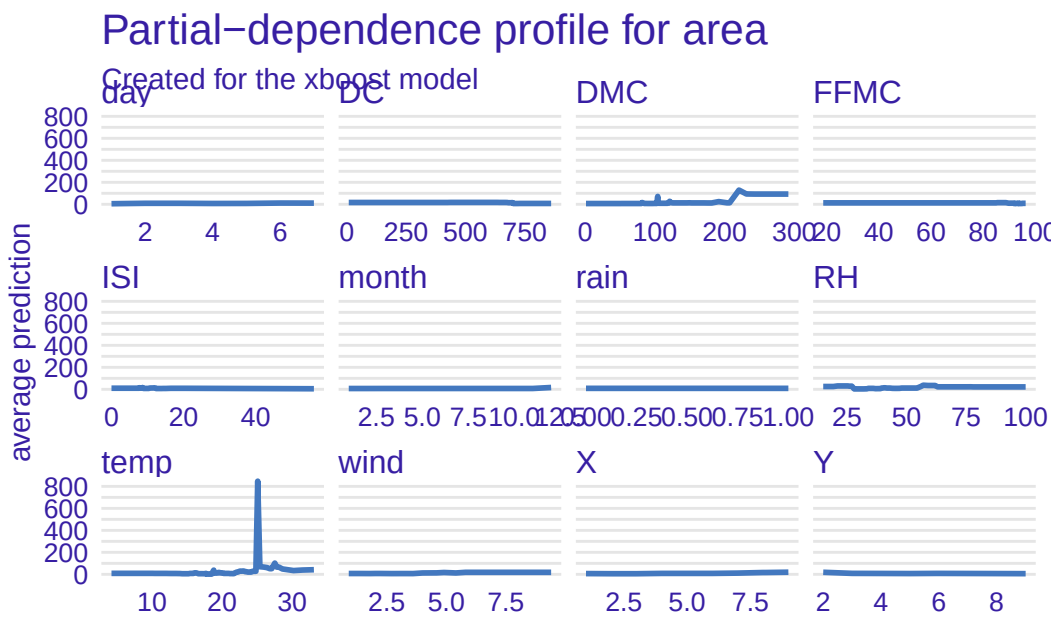
GLOBAL AÇIKLAYICILAR

XBOOST

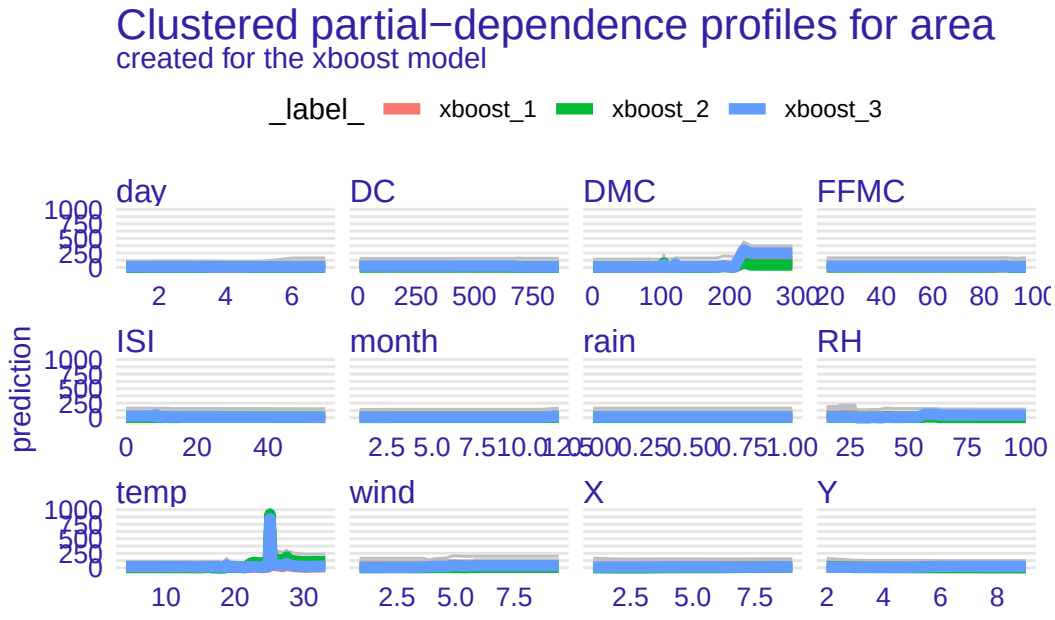
```
library("ggplot2")
# Create variable importance plot
vip_xboost <- model_parts(explainer =explain_xboost_breakdown,
                          loss_function = loss_root_mean_square,
                          B = 50,
                          type = "difference")

# Plot variable importance
library("ggplot2")
plot(vip_xboost ) +
  ggtitle("Mean variable-importance over 50 permutations", "")
```




```
partialvip_xboost <- model_profile(explainer = explain_xboost_breakdown)

library("ggplot2")
plot(partialvip_xboost) +  ggtitle("Partial-dependence profile for area")
```

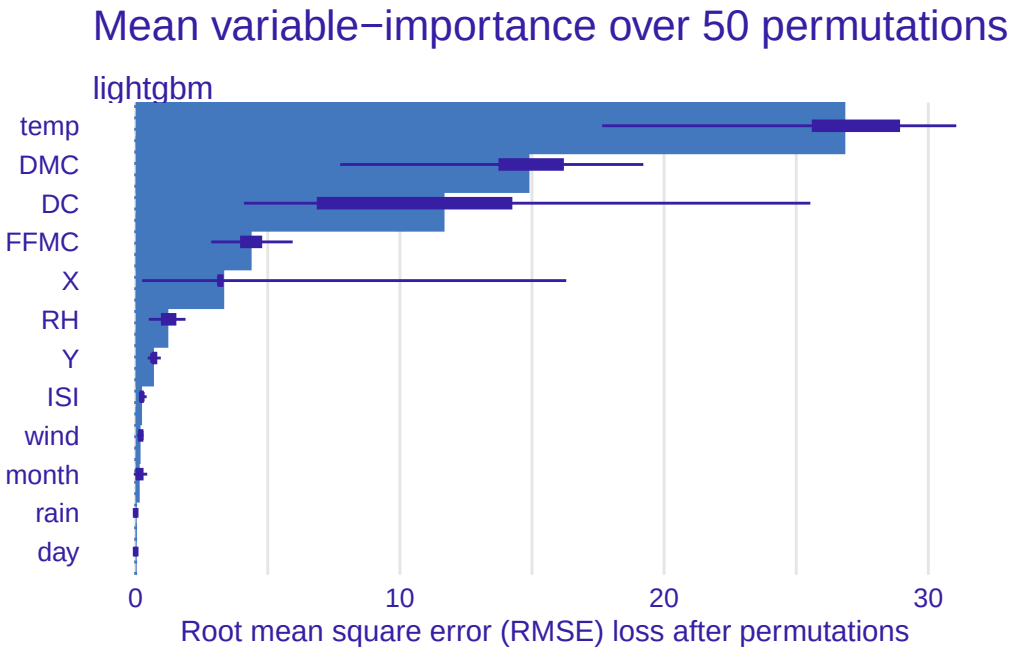



```
pdp_rf_clust <- model_profile(explainer = explain_xboost_breakdown,
                              k = 3)


plot(pdp_rf_clust, geom = "profiles") +
  ggtitle("Clustered partial-dependence profiles for area")
```

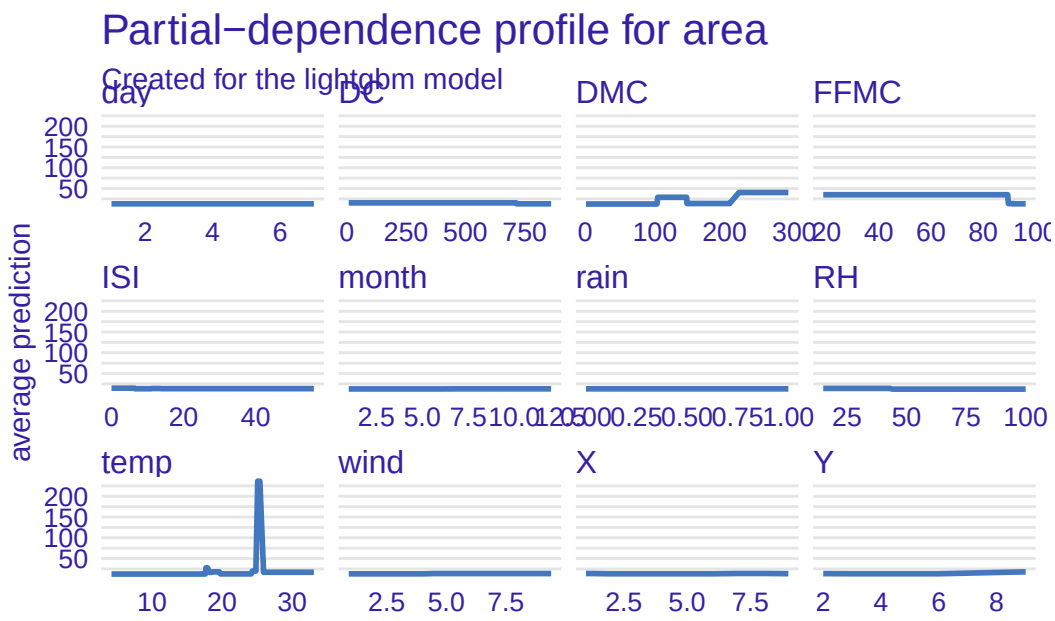


LİGHTGBM

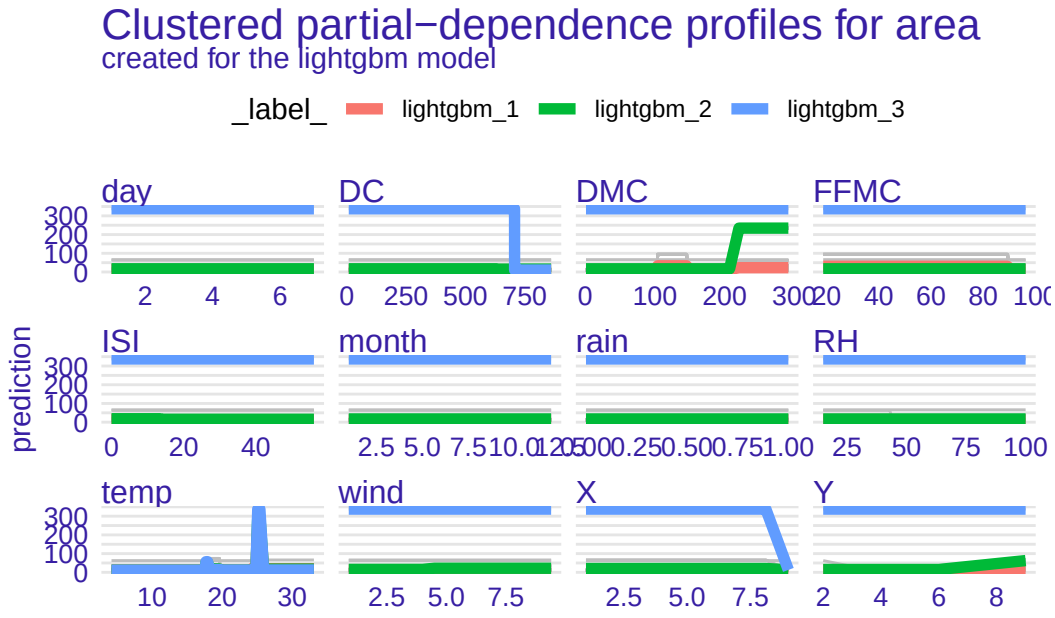
```
vip_lightbm_clust<- model_parts(explainer =explain_lightbm_breakdown ,
                           loss_function = loss_root_mean_square,
                           B = 50,
                           type = "difference")

# Plot variable importance
library("ggplot2")
plot(vip_lightbm_clust ) +
  ggtitle("Mean variable-importance over 50 permutations", "")
```




```
partialvip_lightbm <- model_profile(explainer =explain_lightbm_breakdown )

library("ggplot2")
plot(partialvip_lightbm) +  ggtitle("Partial-dependence profile for area")
```




```
lightbm_clust <- model_profile(explainer = explain_lightbm_breakdown ,
                               k = 3)


plot(lightbm_clust, geom = "profiles") +
  ggtitle("Clustered partial-dependence profiles for area")
```

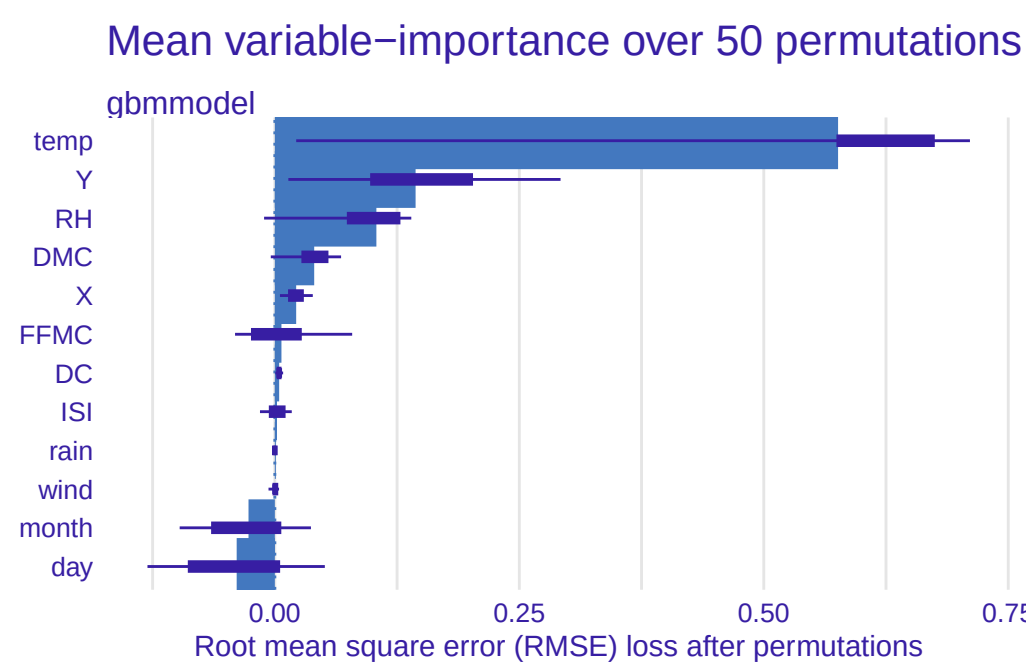


GLOBAL GBMMODEL

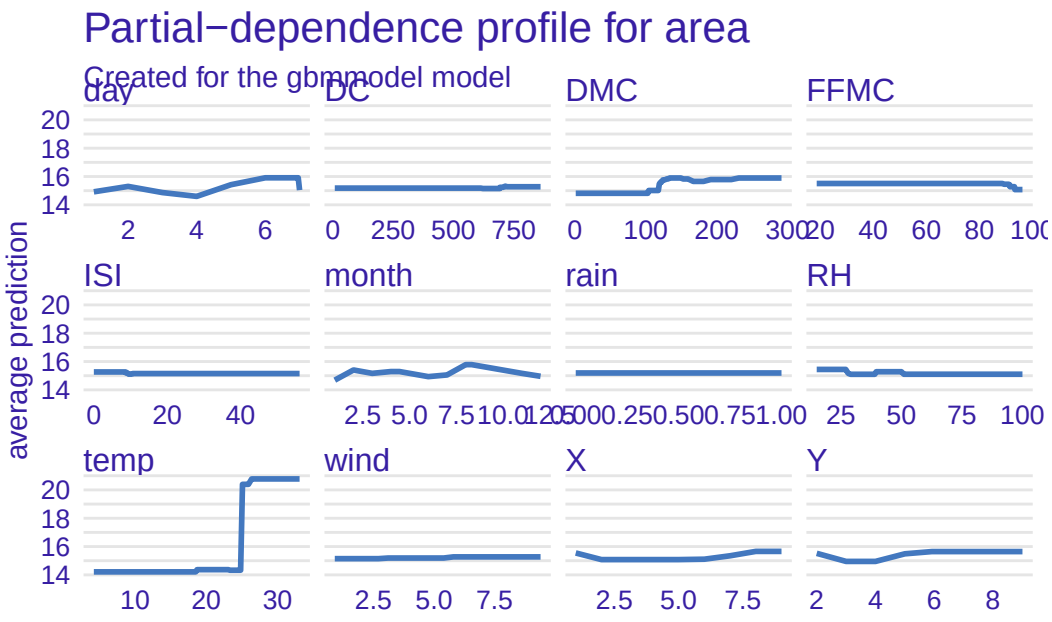
```
explainer_gbmmodel <- DALEX::explain(model = gbmmodel,
                                     data =as.data.frame( train_x),
                                     y = train_y,
                                     is_multiclass = FALSE,
                                     label = "gbmmodel")
```

```
Preparation of a new explainer is initiated
  -> model label       :  gbmmodel
  -> data              :  364  rows  12  cols
  -> target variable   :  364  values
  -> predict function  :  yhat.gbm  will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package gbm , ver. 2.1.8.1 , task regression (  default  )
  -> predicted values  :  numerical, min =  12.97268 , mean =  15.17264 , max =  25.17661
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -24.49661 , mean =  -0.1559598 , max =  1068.442
  A new explainer has been created!
```

```
vip_gbmmodel_clust<- model_parts(explainer =explainer_gbmmodel ,
                                 loss_function = loss_root_mean_square,
                                 B = 50,
                                 type = "difference")

# Plot variable importance
library("ggplot2")
plot(vip_gbmmodel_clust ) +
  ggtitle("Mean variable-importance over 50 permutations", "")
```




```
partialvip_gbmmodel <- model_profile(explainer =explainer_gbmmodel  )

library("ggplot2")
plot(partialvip_gbmmodel) +  ggtitle("Partial-dependence profile for area")
```



```
gbmmodel_clust <- model_profile(explainer = explainer_gbmmodel ,
                                k = 3)


plot(gbmmodel_clust , geom = "profiles") +
  ggtitle("Clustered partial-dependence profiles for area")
```