# Reading the dataset  
# Importing pandas library  
import pandas as pd  
# Loading the dataset  
#lake = pd.read\_csv('E:/XAI/LakeXAI.csv')  
lake = pd.read\_csv('C:/Users/Public/XAI/LakeXAI.csv')

seed = 7

lake

T R\_E R\_AT R\_P G\_E G\_AT G\_P R\_LL \  
0 1/1/2013 0.635469 26.684000 0.060000 12.7 22.55 0.0 281.86   
1 2/1/2013 0.409592 28.214000 0.000000 13.1 24.40 0.0 281.60   
2 3/1/2013 0.501878 30.321000 0.450000 11.5 31.85 0.0 281.34   
3 4/1/2013 1.896789 30.941000 2.530000 11.2 33.15 0.0 281.09   
4 5/1/2013 11.177511 30.767000 8.780000 9.6 32.60 0.0 281.00   
.. ... ... ... ... ... ... ... ...   
103 8/1/2021 54.353870 27.372797 111.480003 2.4 30.80 66.5 281.46   
104 9/1/2021 55.176441 28.473566 85.570000 2.1 30.10 139.3 281.82   
105 10/1/2021 27.200169 30.365717 25.629999 5.8 30.00 0.0 281.99   
106 11/1/2021 4.567200 28.635095 0.000000 8.9 27.45 0.0 281.98   
107 12/1/2021 2.425670 23.681909 0.030000 12.1 23.20 0.0 281.79   
  
 G\_LL   
0 280.39   
1 280.19   
2 279.96   
3 279.65   
4 279.40   
.. ...   
103 279.15   
104 279.53   
105 280.14   
106 280.36   
107 280.28   
  
[108 rows x 9 columns]

lake.head()

T R\_E R\_AT R\_P G\_E G\_AT G\_P R\_LL G\_LL  
0 1/1/2013 0.635469 26.684 0.06 12.7 22.55 0.0 281.86 280.39  
1 2/1/2013 0.409592 28.214 0.00 13.1 24.40 0.0 281.60 280.19  
2 3/1/2013 0.501878 30.321 0.45 11.5 31.85 0.0 281.34 279.96  
3 4/1/2013 1.896789 30.941 2.53 11.2 33.15 0.0 281.09 279.65  
4 5/1/2013 11.177511 30.767 8.78 9.6 32.60 0.0 281.00 279.40

# Exploratory Data Analysis  
# Importing the necessary libraries and functions  
import matplotlib.pyplot as plt  
import seaborn as sns  
%matplotlib inline

# Data description  
lake.describe()

R\_E R\_AT R\_P G\_E G\_AT G\_P \  
count 108.000000 108.000000 108.000000 108.000000 108.000000 108.000000   
mean 20.992243 28.368933 41.858056 9.208333 28.912500 24.374074   
std 24.699415 2.807062 64.048078 4.786432 3.714131 47.042366   
min 0.244746 21.720941 0.000000 0.200000 20.750000 0.000000   
25% 0.789197 26.863723 0.052500 4.975000 25.375000 0.000000   
50% 7.116494 28.687451 4.310000 9.700000 30.275000 0.000000   
75% 44.921698 30.633814 72.317501 13.625000 31.675000 27.075000   
max 75.910866 33.223901 229.970001 16.800000 33.800000 209.700000   
  
 R\_LL G\_LL   
count 108.000000 108.000000   
mean 281.426111 279.761852   
std 0.453857 0.553939   
min 280.510000 278.300000   
25% 281.052500 279.360000   
50% 281.410000 279.745000   
75% 281.822500 280.237500   
max 282.400000 280.830000

### Correlation analysis  
lake\_corr = lake.drop(['T'], axis = 1).corr()  
lake\_corr

R\_E R\_AT R\_P G\_E G\_AT G\_P R\_LL \  
R\_E 1.000000 0.220126 0.914329 -0.895073 0.392979 0.791937 -0.235223   
R\_AT 0.220126 1.000000 0.111853 -0.334988 0.819145 0.017828 -0.519301   
R\_P 0.914329 0.111853 1.000000 -0.795202 0.337752 0.886110 -0.310967   
G\_E -0.895073 -0.334988 -0.795202 1.000000 -0.446201 -0.704554 0.182688   
G\_AT 0.392979 0.819145 0.337752 -0.446201 1.000000 0.233625 -0.607375   
G\_P 0.791937 0.017828 0.886110 -0.704554 0.233625 1.000000 -0.188731   
R\_LL -0.235223 -0.519301 -0.310967 0.182688 -0.607375 -0.188731 1.000000   
G\_LL -0.541670 -0.487631 -0.572185 0.488293 -0.660223 -0.451311 0.857672   
  
 G\_LL   
R\_E -0.541670   
R\_AT -0.487631   
R\_P -0.572185   
G\_E 0.488293   
G\_AT -0.660223   
G\_P -0.451311   
R\_LL 0.857672   
G\_LL 1.000000

lake\_num = lake.drop(['T'], axis = 1)  
lake\_num

R\_E R\_AT R\_P G\_E G\_AT G\_P R\_LL G\_LL  
0 0.635469 26.684000 0.060000 12.7 22.55 0.0 281.86 280.39  
1 0.409592 28.214000 0.000000 13.1 24.40 0.0 281.60 280.19  
2 0.501878 30.321000 0.450000 11.5 31.85 0.0 281.34 279.96  
3 1.896789 30.941000 2.530000 11.2 33.15 0.0 281.09 279.65  
4 11.177511 30.767000 8.780000 9.6 32.60 0.0 281.00 279.40  
.. ... ... ... ... ... ... ... ...  
103 54.353870 27.372797 111.480003 2.4 30.80 66.5 281.46 279.15  
104 55.176441 28.473566 85.570000 2.1 30.10 139.3 281.82 279.53  
105 27.200169 30.365717 25.629999 5.8 30.00 0.0 281.99 280.14  
106 4.567200 28.635095 0.000000 8.9 27.45 0.0 281.98 280.36  
107 2.425670 23.681909 0.030000 12.1 23.20 0.0 281.79 280.28  
  
[108 rows x 8 columns]

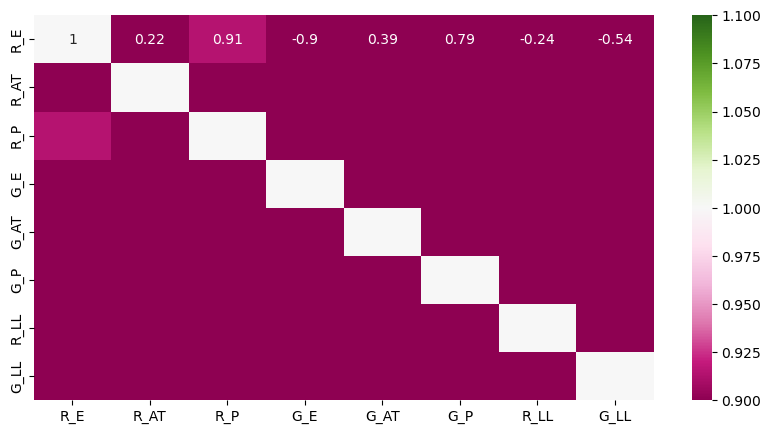
lake\_corr['R\_LL']

R\_E -0.235223  
R\_AT -0.519301  
R\_P -0.310967  
G\_E 0.182688  
G\_AT -0.607375  
G\_P -0.188731  
R\_LL 1.000000  
G\_LL 0.857672  
Name: R\_LL, dtype: float64

lake\_corr['G\_LL']

R\_E -0.541670  
R\_AT -0.487631  
R\_P -0.572185  
G\_E 0.488293  
G\_AT -0.660223  
G\_P -0.451311  
R\_LL 0.857672  
G\_LL 1.000000  
Name: G\_LL, dtype: float64

# Correlation coefficients using SNS heatmap  
plt.figure(figsize=(10, 5))  
sns.heatmap(lake\_corr, vmin = 1, vmax = 1, cmap = 'PiYG', annot = True)  
plt.show()



# Important features for each target

# R\_LL  
RLL\_corr = lake\_corr['R\_LL']  
RLL\_corr

R\_E -0.235223  
R\_AT -0.519301  
R\_P -0.310967  
G\_E 0.182688  
G\_AT -0.607375  
G\_P -0.188731  
R\_LL 1.000000  
G\_LL 0.857672  
Name: R\_LL, dtype: float64

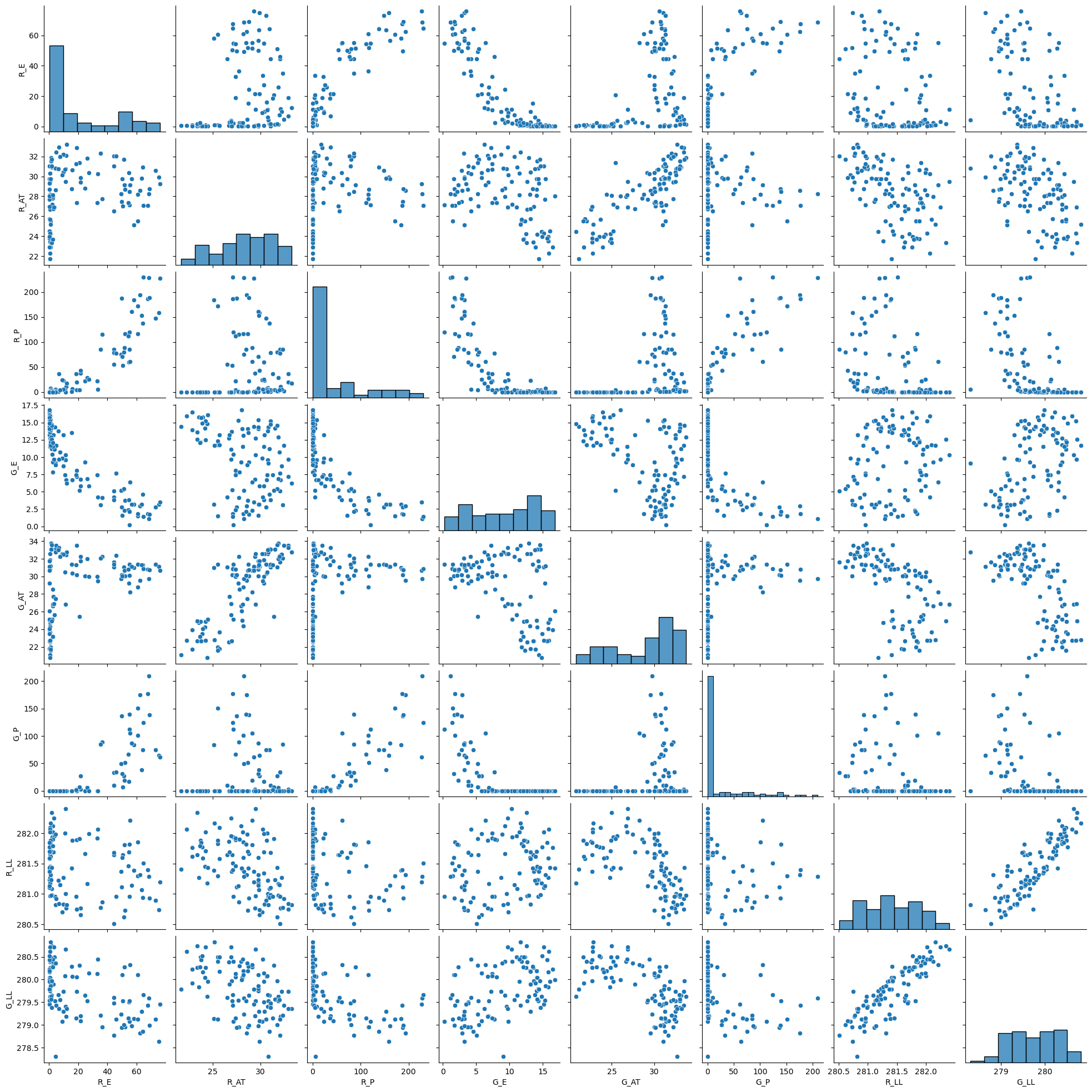
GLL\_corr = lake\_corr['G\_LL']  
GLL\_corr

R\_E -0.541670  
R\_AT -0.487631  
R\_P -0.572185  
G\_E 0.488293  
G\_AT -0.660223  
G\_P -0.451311  
R\_LL 0.857672  
G\_LL 1.000000  
Name: G\_LL, dtype: float64

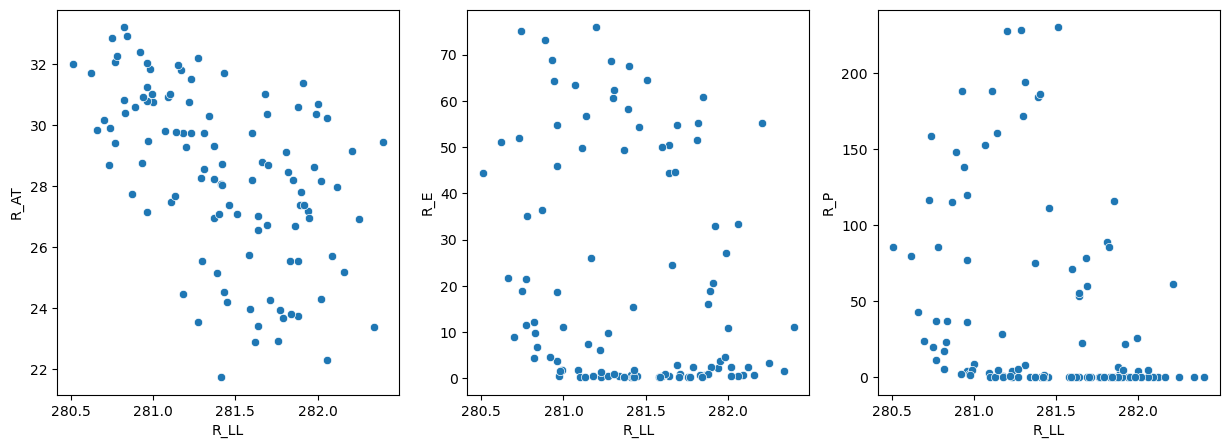
# Visualization

sns.pairplot(lake\_num)

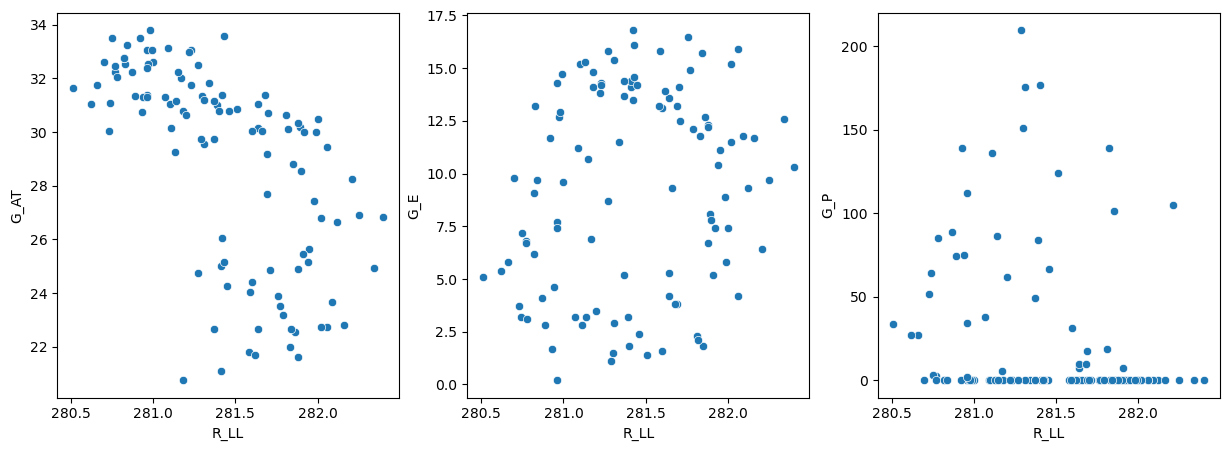
<seaborn.axisgrid.PairGrid at 0x25d86554730>



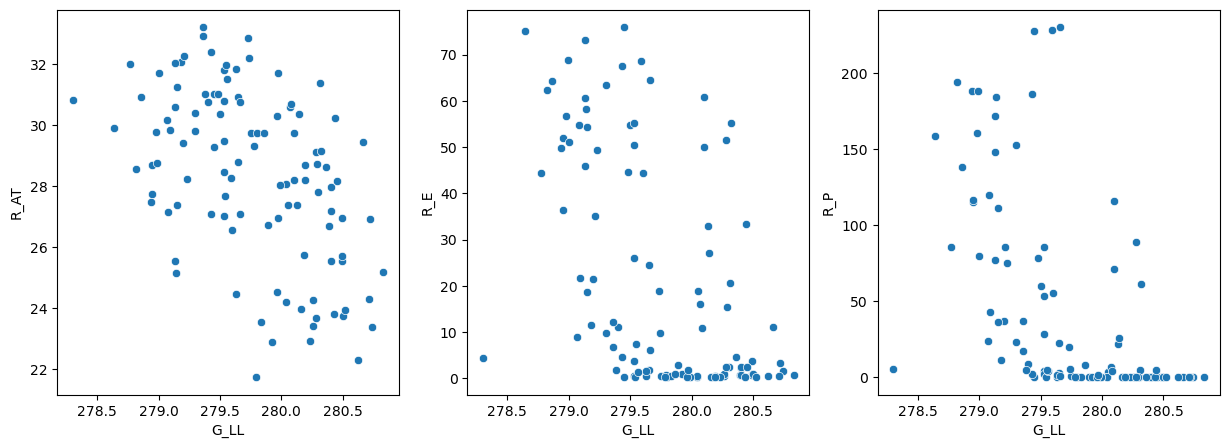
# R\_LL vs RS features  
fig, axes = plt.subplots(1,3, sharex = True, figsize = (15,5))  
sns.scatterplot(data = lake, x = 'R\_LL', y = 'R\_AT', ax = axes[0])  
sns.scatterplot(data = lake, x = 'R\_LL', y = 'R\_E', ax = axes[1])  
sns.scatterplot(data = lake, x = 'R\_LL', y = 'R\_P', ax = axes[2])  
plt.show()



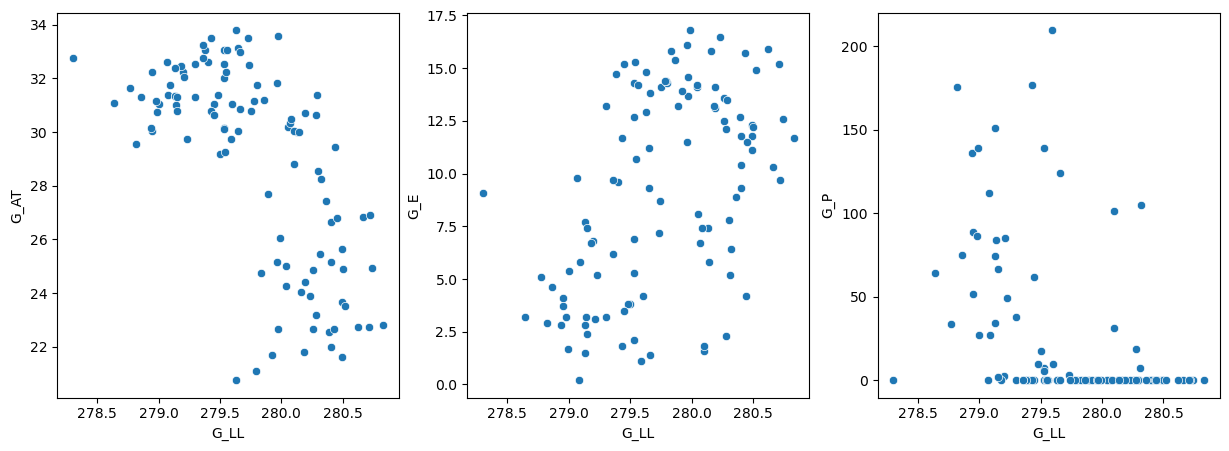
# R\_LL vs GT features  
fig, axes = plt.subplots(1,3, sharex = True, figsize = (15,5))  
sns.scatterplot(data = lake, x = 'R\_LL', y = 'G\_AT', ax = axes[0])  
sns.scatterplot(data = lake, x = 'R\_LL', y = 'G\_E', ax = axes[1])  
sns.scatterplot(data = lake, x = 'R\_LL', y = 'G\_P', ax = axes[2])  
plt.show()



# G\_LL vs RS features  
fig, axes = plt.subplots(1,3, sharex = True, figsize = (15,5))  
sns.scatterplot(data = lake, x = 'G\_LL', y = 'R\_AT', ax = axes[0])  
sns.scatterplot(data = lake, x = 'G\_LL', y = 'R\_E', ax = axes[1])  
sns.scatterplot(data = lake, x = 'G\_LL', y = 'R\_P', ax = axes[2])  
plt.show()

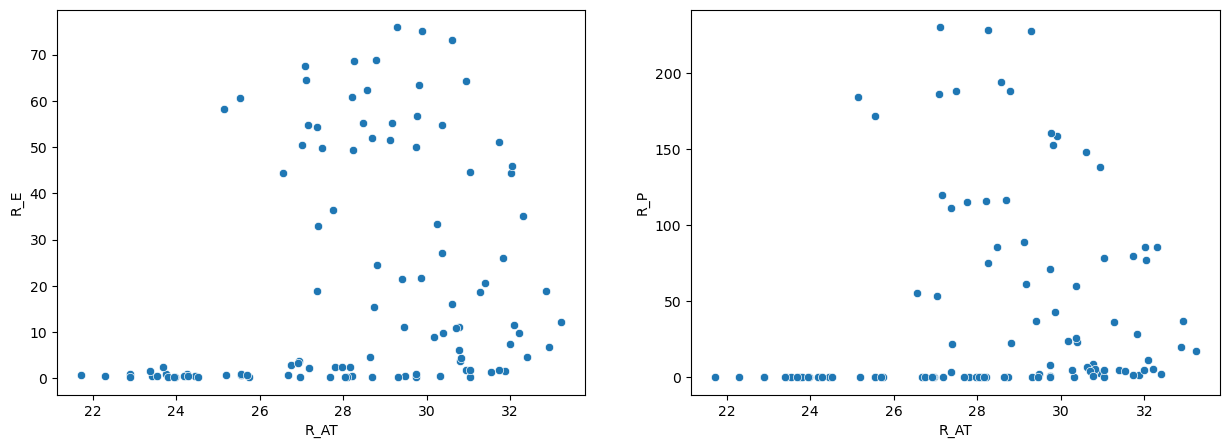


# LL\_G vs GT features  
fig, axes = plt.subplots(1,3, sharex = True, figsize = (15,5))  
sns.scatterplot(data = lake, x = 'G\_LL', y = 'G\_AT', ax = axes[0])  
sns.scatterplot(data = lake, x = 'G\_LL', y = 'G\_E', ax = axes[1])  
sns.scatterplot(data = lake, x = 'G\_LL', y = 'G\_P', ax = axes[2])  
plt.show()

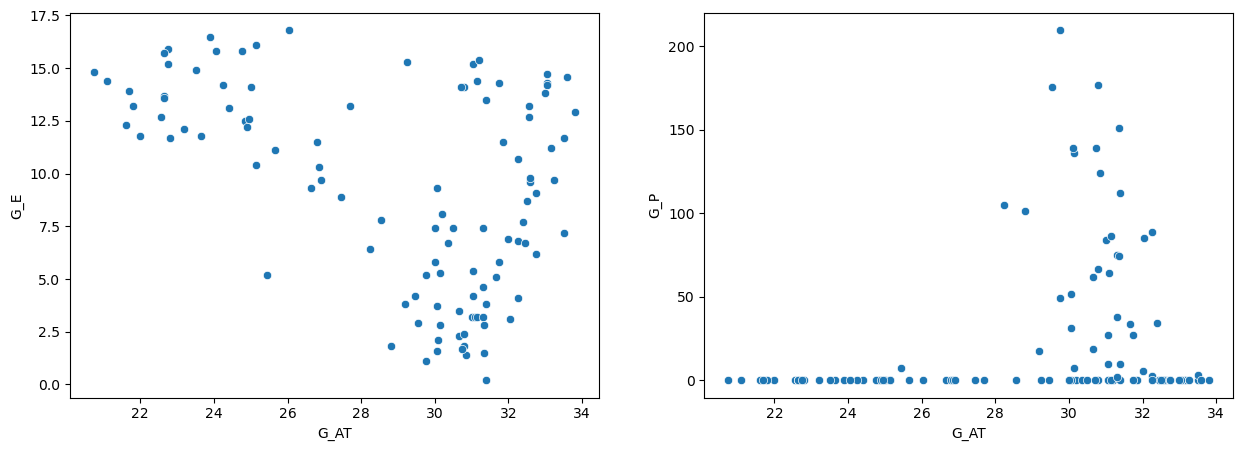


# Features vs Features

# RS  
fig, axes = plt.subplots(1,2, sharex = True, figsize = (15,5))  
sns.scatterplot(data = lake, x = 'R\_AT', y = 'R\_E', ax = axes[0])  
sns.scatterplot(data = lake, x = 'R\_AT', y = 'R\_P', ax = axes[1])  
plt.show()

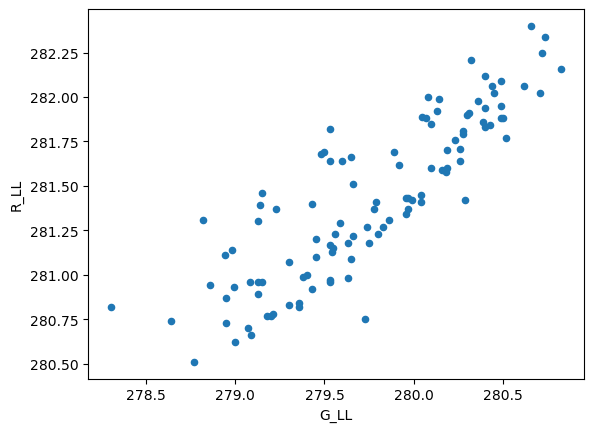


# GT  
fig, axes = plt.subplots(1,2, sharex = True, figsize = (15,5))  
sns.scatterplot(data = lake, x = 'G\_AT', y = 'G\_E', ax = axes[0])  
sns.scatterplot(data = lake, x = 'G\_AT', y = 'G\_P', ax = axes[1])  
plt.show()



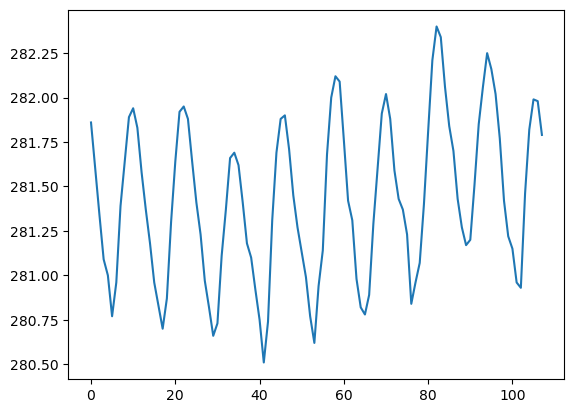
lake.plot.scatter(x = 'G\_LL', y = 'R\_LL')

<Axes: xlabel='G\_LL', ylabel='R\_LL'>



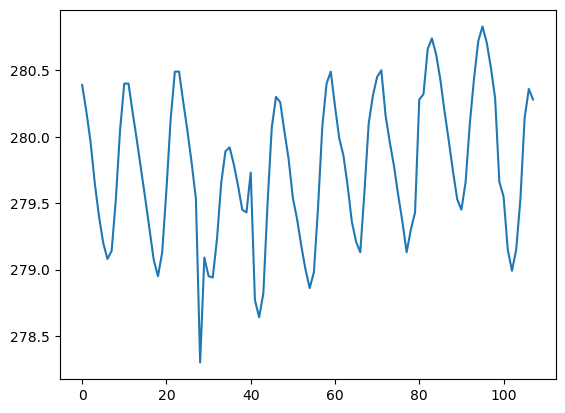
lake['R\_LL'].plot()

<Axes: >



lake['G\_LL'].plot()

<Axes: >



## Preparing the input and output variables  
  
# import numpy as np

# The output features

# The remote sensing ouput target  
y\_R = lake['R\_LL']

y\_R.shape

(108,)

print(y\_R[:5])

0 281.86  
1 281.60  
2 281.34  
3 281.09  
4 281.00  
Name: R\_LL, dtype: float64

# The ground ouput target  
y\_G = lake['G\_LL']

y\_G.shape

(108,)

print(y\_G[:5])

0 280.39  
1 280.19  
2 279.96  
3 279.65  
4 279.40  
Name: G\_LL, dtype: float64

# The input features

# The remote sensing input features  
  
  
X\_R = lake.drop(['T','R\_LL','G\_LL','G\_E', 'G\_AT','G\_P'], axis = 1)

X\_R.shape

(108, 3)

print(X\_R[:5])

R\_E R\_AT R\_P  
0 0.635469 26.684 0.06  
1 0.409592 28.214 0.00  
2 0.501878 30.321 0.45  
3 1.896789 30.941 2.53  
4 11.177511 30.767 8.78

# The ground input features  
  
X\_G = lake.drop(['T','R\_LL','G\_LL','R\_E', 'R\_AT','R\_P'], axis = 1)

X\_G.shape

(108, 3)

print(X\_G[:5])

G\_E G\_AT G\_P  
0 12.7 22.55 0.0  
1 13.1 24.40 0.0  
2 11.5 31.85 0.0  
3 11.2 33.15 0.0  
4 9.6 32.60 0.0

# Standardizing / Scaling the features  
# Importing the StandardScaler function  
from sklearn.preprocessing import StandardScaler

# Scaling the RS input features  
X\_R\_scaled = pd.DataFrame(StandardScaler().fit\_transform(X\_R),  
columns= X\_R.columns, index = X\_R.index)

# Scaling the GT input features  
X\_G\_scaled = pd.DataFrame(StandardScaler().fit\_transform(X\_G),  
columns= X\_G.columns, index = X\_G.index)

# VIF  
from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

### X\_R vIF  
vif = pd.DataFrame()  
vif['vif'] = [variance\_inflation\_factor(X\_R\_scaled,i) for i in range(X\_R\_scaled.shape[1])]  
vif['Features'] = X\_R.columns  
vif.round(2)

vif Features  
0 6.67 R\_E  
1 1.11 R\_AT  
2 6.43 R\_P

### X\_G VIF  
vif = pd.DataFrame()  
vif['vif'] = [variance\_inflation\_factor(X\_G\_scaled,i) for i in range(X\_G\_scaled.shape[1])]  
vif['Features'] = X\_G.columns  
vif.round(2)

vif Features  
0 2.38 G\_E  
1 1.27 G\_AT  
2 2.02 G\_P

# Splitting the feature data into training and testing datasets  
# Importing train\_test\_splt function  
from sklearn.model\_selection import train\_test\_split

# Splitting the RS Data  
  
X\_R\_train, X\_R\_test, y\_R\_train, y\_R\_test = train\_test\_split(X\_R\_scaled, y\_R, test\_size = 0.25, random\_state = 30, shuffle = True)

print(X\_R\_train.shape, X\_R\_test.shape)

(81, 3) (27, 3)

# Splitting the GT Data  
  
X\_G\_train, X\_G\_test, y\_G\_train, y\_G\_test = train\_test\_split(X\_G\_scaled, y\_G, test\_size = 0.25, random\_state = 30, shuffle = True)

print(X\_G\_train.shape, X\_G\_test.shape)

(81, 3) (27, 3)

######### LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS (LIME) ####################

####### Installing LIME by typing !pip install lime command ########  
# !pip install lime or conda install conda-forge::xgboost

### Importing the necessary libraries #########  
import pandas as pd  
import numpy as np  
from numpy import absolute  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score  
from lime.lime\_tabular import LimeTabularExplainer  
from sklearn.model\_selection import cross\_val\_score, KFold, GridSearchCV

# Importing wanrnings to ignore warning messages  
import warnings  
warnings.filterwarnings("ignore")

############################### LINEAR REGRESSION ###############################

####### MODEL 1: RS Features (X\_R) and RS Target (LL\_R)#####################

# Instantiation using LinearRegression  
model = LinearRegression(fit\_intercept = True, copy\_X = True, positive = False, n\_jobs = None)

###### Training and fiting the linear model on the whole dataset  
lm1 = model.fit(X\_R, y\_R)

# The model interept  
print('The intercept is:', lm1.intercept\_)

The intercept is: 284.0585251148753

# The model coefficients  
print('The model coefficients are:', lm1.coef\_)

The model coefficients are: [ 0.01310467 -0.09309344 -0.00636792]

# The coefficient of determination  
  
print('The coefficient of determination is:', lm1.score(X\_R, y\_R))

The coefficient of determination is: 0.4106927221973462

# Finding metrics on the whole dataset

# Making the prediction  
lm1\_pred = lm1.predict(X\_R)

print('The R2 is:', r2\_score(y\_R, lm1\_pred))

The R2 is: 0.4106927221973462

# lm1 MAE  
print('The lm1 MAE is:%.2f'% mean\_absolute\_error(y\_R, lm1\_pred))

The lm1 MAE is:0.28

# lm1 MSE  
print('The lm1 MSE is:%.2f'% mean\_squared\_error(y\_R, lm1\_pred))

The lm1 MSE is:0.12

# OR  
lm1\_pred = lm1.predict(X\_R)  
print('The R2 is:', lm1.score(X\_R, y\_R))  
print('The MAE is:', mean\_absolute\_error(y\_R, lm1\_pred))  
print('The MSE is:', mean\_squared\_error(y\_R, lm1\_pred))

The R2 is: 0.4106927221973462  
The MAE is: 0.28387040423882737  
The MSE is: 0.1202654291070482

## The k-fold Cross-validation  
import numpy as np

# On the training dataset  
score\_lm1 = cross\_val\_score(lm1, X\_R, y\_R, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_lm1

array([-0.10321261, -0.0770486 , -0.12810855, -0.18359509, -0.16105948,  
 -0.1294216 , -0.1043502 , -0.18241232, -0.17942713, -0.07903033])

# The absolute mean score  
from numpy import absolute

print('The CV MSE is:', absolute(np.mean(score\_lm1)))

The CV MSE is: 0.13276659156087695

# LIME

###### Initializing the explainer on the whole features dataset  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_R\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the full datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, lm1.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 281.32393123781566  
Prediction\_local [281.83769739]  
Right: 281.5823653670459  
Intercept 281.4561043542672  
Prediction\_local [281.48477548]  
Right: 281.43735444722466  
Intercept 281.4833464387461  
Prediction\_local [281.36518133]  
Right: 281.239550388344  
Intercept 281.4121367310977  
Prediction\_local [281.1666921]  
Right: 281.1868670124363  
Intercept 281.6799578850307  
Prediction\_local [281.26950707]  
Right: 281.2848865323185  
Intercept 281.3331111012729  
Prediction\_local [281.47271642]  
Right: 281.3652395056614  
Intercept 281.456765025763  
Prediction\_local [281.35315052]  
Right: 281.4865446992454  
Intercept 281.3437539845363  
Prediction\_local [281.61756628]  
Right: 281.30635198262434  
Intercept 281.1254538948989  
Prediction\_local [281.97608091]  
Right: 281.8658722217656  
Intercept 281.31026122230145  
Prediction\_local [281.90172126]  
Right: 281.73467009258866  
Intercept 281.4697176191778  
Prediction\_local [281.45998285]  
Right: 281.5573316629188  
Intercept 281.2897256750402  
Prediction\_local [281.78795539]  
Right: 281.69056459513183  
Intercept 281.3484991115534  
Prediction\_local [281.76908981]  
Right: 281.6666469566413  
Intercept 281.46272074163437  
Prediction\_local [281.47674396]  
Right: 281.5523911447239  
Intercept 281.3871937223238  
Prediction\_local [281.43656036]  
Right: 281.290560759421  
Intercept 281.5294553108881  
Prediction\_local [281.15395386]  
Right: 281.2186577010367  
Intercept 281.2850187878778  
Prediction\_local [281.47595136]  
Right: 281.21199407798434  
Intercept 281.46144446159593  
Prediction\_local [281.25986997]  
Right: 281.21555926033096  
Intercept 281.5807138312685  
Prediction\_local [281.00098918]  
Right: 281.21706017086314  
Intercept 281.2530993532631  
Prediction\_local [281.65915759]  
Right: 281.3845581137325  
Intercept 281.40187033230313  
Prediction\_local [281.74062333]  
Right: 281.8172307646224  
Intercept 281.22114581184763  
Prediction\_local [281.70058144]  
Right: 281.7996786197332  
Intercept 281.40380401922613  
Prediction\_local [281.48951173]  
Right: 281.60008468378186  
Intercept 281.22513902215604  
Prediction\_local [281.5780834]  
Right: 281.69303641998596  
Intercept 281.401312361178  
Prediction\_local [281.76525926]  
Right: 281.88433442705485  
Intercept 281.3382822419194  
Prediction\_local [281.60524255]  
Right: 281.4482692545888  
Intercept 281.3531720303914  
Prediction\_local [281.26978598]  
Right: 281.29017131112045  
Intercept 281.3515895837877  
Prediction\_local [281.31418946]  
Right: 281.3110818252592  
Intercept 281.47300995941754  
Prediction\_local [280.82520894]  
Right: 281.2124408927923  
Intercept 281.42933451436676  
Prediction\_local [281.49473828]  
Right: 281.2887432788572  
Intercept 281.454035473279  
Prediction\_local [281.34453292]  
Right: 281.32648183019893  
Intercept 281.4407415607536  
Prediction\_local [281.31060302]  
Right: 280.9536035281034  
Intercept 281.44990111669534  
Prediction\_local [281.30156532]  
Right: 281.59669401631413  
Intercept 281.5673031305531  
Prediction\_local [281.44879527]  
Right: 281.5564088161969  
Intercept 281.37064383845956  
Prediction\_local [281.731295]  
Right: 281.6059207433613  
Intercept 281.24098686695555  
Prediction\_local [281.76834969]  
Right: 281.9389673248908  
Intercept 281.25940412547925  
Prediction\_local [281.79746801]  
Right: 282.04444979006496  
Intercept 281.3248765548271  
Prediction\_local [281.6117217]  
Right: 281.78843559389884  
Intercept 281.4583480143301  
Prediction\_local [281.06572377]  
Right: 281.17125781264264  
Intercept 281.3767369035843  
Prediction\_local [281.33571426]  
Right: 281.0877767030936  
Intercept 281.48556366513225  
Prediction\_local [281.13571259]  
Right: 281.12109364732447  
Intercept 281.7518804664189  
Prediction\_local [280.51098589]  
Right: 281.11474640582827  
Intercept 281.52879156065234  
Prediction\_local [281.11673139]  
Right: 281.2512232091498  
Intercept 281.4282168654459  
Prediction\_local [281.10271973]  
Right: 280.9817520008466  
Intercept 281.2638193291377  
Prediction\_local [281.7787613]  
Right: 281.5690672020233  
Intercept 281.5086704682251  
Prediction\_local [281.35762301]  
Right: 281.3749044625612  
Intercept 281.22976712591134  
Prediction\_local [281.57778467]  
Right: 281.50172434757025  
Intercept 281.2641175417176  
Prediction\_local [281.86185927]  
Right: 281.8106723163543  
Intercept 281.2885308236052  
Prediction\_local [281.91539052]  
Right: 281.81184332064873  
Intercept 281.2859099383754  
Prediction\_local [281.7028954]  
Right: 281.8724762784322  
Intercept 281.4191803196762  
Prediction\_local [281.31010989]  
Right: 281.48288697620785  
Intercept 281.6437819867789  
Prediction\_local [281.14078673]  
Right: 281.16375503693115  
Intercept 281.4865650430864  
Prediction\_local [281.21646868]  
Right: 281.1500020602075  
Intercept 281.53628845632255  
Prediction\_local [281.04712694]  
Right: 281.2688652621391  
Intercept 281.56360125128793  
Prediction\_local [280.98448489]  
Right: 281.1428094518713  
Intercept 281.59149473308  
Prediction\_local [281.15383501]  
Right: 281.00560803186994  
Intercept 281.71721015292667  
Prediction\_local [280.62072017]  
Right: 281.25525363380245  
Intercept 281.44937430063624  
Prediction\_local [281.38677689]  
Right: 281.3193343085261  
Intercept 281.50835505305275  
Prediction\_local [281.52526946]  
Right: 281.4866108884286  
Intercept 281.2654271002461  
Prediction\_local [281.82839379]  
Right: 281.67566802898034  
Intercept 281.1858986513656  
Prediction\_local [281.79840337]  
Right: 281.9306980160054  
Intercept 281.40497080858677  
Prediction\_local [281.46093587]  
Right: 281.4512994191485  
Intercept 281.5447432217557  
Prediction\_local [281.29487542]  
Right: 281.24891792802424  
Intercept 281.5493747902577  
Prediction\_local [281.03643958]  
Right: 281.10171882839455  
Intercept 281.3213001847169  
Prediction\_local [281.2836447]  
Right: 281.0169911474261  
Intercept 281.71396223813616  
Prediction\_local [280.49008871]  
Right: 280.96749297784294  
Intercept 281.49752612341223  
Prediction\_local [281.13918637]  
Right: 281.2256747636486  
Intercept 281.52435088625714  
Prediction\_local [281.24113658]  
Right: 280.87278260481287  
Intercept 281.37692274050585  
Prediction\_local [281.91601487]  
Right: 281.4902570451322  
Intercept 281.48065597693386  
Prediction\_local [281.15286966]  
Right: 281.3778764794391  
Intercept 281.4098300069145  
Prediction\_local [281.50278831]  
Right: 281.4697969318329  
Intercept 281.3079567633137  
Prediction\_local [281.80812767]  
Right: 281.8584269498268  
Intercept 281.26010414712744  
Prediction\_local [281.82334166]  
Right: 281.832238095236  
Intercept 281.4635017391671  
Prediction\_local [281.79955002]  
Right: 281.77993239388604  
Intercept 281.4979399213503  
Prediction\_local [281.20290743]  
Right: 281.3304470346736  
Intercept 281.4380021174781  
Prediction\_local [281.27138871]  
Right: 281.11397676263124  
Intercept 281.70305911475305  
Prediction\_local [280.79764704]  
Right: 280.84915934008507  
Intercept 281.548496443295  
Prediction\_local [281.03228883]  
Right: 281.18783095196744  
Intercept 281.5053344763539  
Prediction\_local [281.21069321]  
Right: 281.14184774092575  
Intercept 281.4885748084401  
Prediction\_local [281.36491278]  
Right: 281.2362808784081  
Intercept 281.5188621127321  
Prediction\_local [281.25221399]  
Right: 281.4579234999545  
Intercept 281.20977129668125  
Prediction\_local [281.89049738]  
Right: 281.6808352556936  
Intercept 281.37282920903436  
Prediction\_local [281.56298279]  
Right: 281.4631359183687  
Intercept 281.3351912281597  
Prediction\_local [281.7895042]  
Right: 281.9021073376198  
Intercept 281.28186349255327  
Prediction\_local [281.73449479]  
Right: 281.9914261809664  
Intercept 281.1290684396079  
Prediction\_local [281.93744722]  
Right: 281.8476391645994  
Intercept 281.4125675586933  
Prediction\_local [281.31389821]  
Right: 281.38933228269366  
Intercept 281.4591188903595  
Prediction\_local [281.06246306]  
Right: 281.1203252488214  
Intercept 281.6006873665141  
Prediction\_local [281.34895914]  
Right: 281.1543994644047  
Intercept 281.5096890207135  
Prediction\_local [281.22382135]  
Right: 281.2549091030529  
Intercept 281.49026368750435  
Prediction\_local [281.20660263]  
Right: 280.8807840431887  
Intercept 281.4301061944396  
Prediction\_local [281.26094211]  
Right: 280.9163548458163  
Intercept 281.40688189738967  
Prediction\_local [281.37697846]  
Right: 281.49211973103905  
Intercept 281.41474476882087  
Prediction\_local [281.28173723]  
Right: 281.6498031304441  
Intercept 281.3801948266584  
Prediction\_local [281.37365892]  
Right: 281.59631143021625  
Intercept 281.23527199109395  
Prediction\_local [281.85589396]  
Right: 281.7247440460222  
Intercept 281.43432388817837  
Prediction\_local [281.78108885]  
Right: 281.80352851684137  
Intercept 281.1548673757398  
Prediction\_local [281.71235532]  
Right: 281.83282644199437  
Intercept 281.4027062702574  
Prediction\_local [281.42975153]  
Right: 281.58252685222993  
Intercept 281.5269472441121  
Prediction\_local [281.06860261]  
Right: 281.2705526897869  
Intercept 281.4301472402221  
Prediction\_local [281.11083621]  
Right: 281.14871885579134  
Intercept 281.45709315495594  
Prediction\_local [281.20244319]  
Right: 281.1595295477687  
Intercept 281.4242234456783  
Prediction\_local [281.2592904]  
Right: 281.0844053681005  
Intercept 281.3654922840853  
Prediction\_local [281.5285968]  
Right: 281.5126906910678  
Intercept 281.44146197303814  
Prediction\_local [281.47978507]  
Right: 281.58598873458726  
Intercept 281.466540913546  
Prediction\_local [281.45376559]  
Right: 281.4249154732121  
Intercept 281.4810771475858  
Prediction\_local [281.32104885]  
Right: 281.45263730899234  
Intercept 281.3430555714858  
Prediction\_local [281.86961181]  
Right: 281.8854913490758

exp

<lime.explanation.Explanation at 0x1d5fb1076a0>

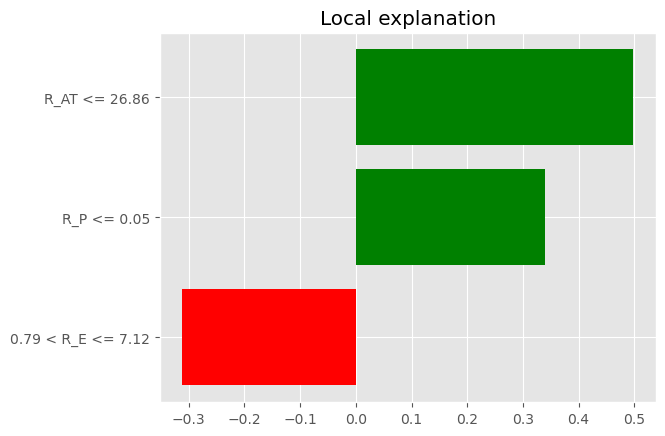
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_R.columns);  
 plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_AT <= 26.86', 0.49795238367186184),  
 ('R\_P <= 0.05', 0.3402125434553615),  
 ('0.79 < R\_E <= 7.12', -0.3116086860657837)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.49795238367186184),  
 (2, -0.3402125434553615),  
 (0, 0.3116086860657837)],  
 1: [(1, 0.49795238367186184),  
 (2, 0.3402125434553615),  
 (0, -0.3116086860657837)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.86961181]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.8854913490758

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_RR\_Whole.html')

##################################################################################################

################## Training and fiting the linear model on the training dataset #####################  
lm1 = model.fit(X\_R\_train, y\_R\_train)

# The training data metrics

#### Predicting on the training dataset  
lm1\_predtr = lm1.predict(X\_R\_train)

# The R-sq on the trainong dataset  
  
print('The training R2 is:', lm1.score(X\_R\_train, y\_R\_train))

The training R2 is: 0.44303733210838914

# The training MAE  
mae\_tr = mean\_absolute\_error(y\_R\_train, lm1\_predtr)

print('The training MAE is:', mae\_tr)

The training MAE is: 0.27675357215213897

# The training MSE  
mse\_tr = mean\_squared\_error(y\_R\_train, lm1\_predtr)

print('The training MSE is:', mse\_tr)

The training MSE is: 0.1176655070776237

# OR  
lm1\_predtr = lm1.predict(X\_R\_train)  
print('The Training R2 is:', lm1.score(X\_R\_train, y\_R\_train))  
print('The training MAE is:', mean\_absolute\_error(y\_R\_train, lm1\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_R\_train, lm1\_predtr))

The Training R2 is: 0.44303733210838914  
The training MAE is: 0.27675357215213897  
The training MSE is: 0.1176655070776237

####### Predicting on the testing dataset  
  
lm1\_predts = lm1.predict(X\_R\_test)

# The testing data metrics

# The R-sq on the testing dataset  
print('The testing R2 is:', lm1.score(X\_R\_test, y\_R\_test))

The testing R2 is: 8.689710867582079e-05

# The testing MAE  
mae\_ts = mean\_absolute\_error(y\_R\_test, lm1\_predts)

print('The testing MAE is:', mae\_ts)

The testing MAE is: 0.3050353231088905

##### The Testing MSE  
mse\_ts = mean\_squared\_error(y\_R\_test, lm1\_predts)

#### Printing the MSE  
print(' The Testing MSE is:', mse\_ts)

The Testing MSE is: 0.1452318231221702

# OR  
lm1\_predts = lm1.predict(X\_R\_test)  
print('The testing R2 is:', lm1.score(X\_R\_test, y\_R\_test))  
print('The testing MAE is:', mean\_absolute\_error(y\_R\_test, lm1\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_R\_test, lm1\_predts))

The testing R2 is: 8.689710867582079e-05  
The testing MAE is: 0.3050353231088905  
The testing MSE is: 0.1452318231221702

## The k-fold Cross-validation

# On the training dataset  
score\_train = cross\_val\_score(lm1, X\_R\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_train

array([-0.17818671, -0.04128197, -0.11275759, -0.15568741, -0.20288849,  
 -0.16493012, -0.05192244, -0.07607289, -0.14739403, -0.18355256])

# The absolute mean score on the training dataset

print('The training CV MSE is:', absolute(np.mean(score\_train)))

The training CV MSE is: 0.1314674193945717

## On the testing dataset  
score\_test = cross\_val\_score(lm1, X\_R\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

print('The testing CV MSE is:', absolute(np.mean(score\_test)))

The testing CV MSE is: 0.11949448140737921

# Explain individual predictionsusing LimeTabularExplainer

# 1. Creating Explainer object

###### Initializing the explainer on the training dataset  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_R\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, lm1.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.2678598140942  
Prediction\_local [281.25772099]  
Right: 281.2774304398639  
Intercept 281.4425966616251  
Prediction\_local [281.03246552]  
Right: 281.2278860160087  
Intercept 281.2127952599988  
Prediction\_local [281.85062984]  
Right: 281.65375101745815  
Intercept 281.4385928527998  
Prediction\_local [281.08866766]  
Right: 281.19115515906094  
Intercept 281.39827932181583  
Prediction\_local [280.99593879]  
Right: 280.761570189788  
Intercept 281.19600804025015  
Prediction\_local [281.48376856]  
Right: 281.52952207717914  
Intercept 281.52570156303665  
Prediction\_local [281.0855103]  
Right: 280.702955843282  
Intercept 281.5160176313149  
Prediction\_local [281.19226351]  
Right: 281.24676844866036  
Intercept 281.2837251881669  
Prediction\_local [281.29588969]  
Right: 281.09793065702297  
Intercept 281.567512914692  
Prediction\_local [281.27348845]  
Right: 281.048005077694  
Intercept 281.54431963250255  
Prediction\_local [281.21361384]  
Right: 281.44097802169955  
Intercept 281.43036515015405  
Prediction\_local [281.22803076]  
Right: 281.031044751857  
Intercept 281.435370672752  
Prediction\_local [281.56062663]  
Right: 281.341598278932  
Intercept 281.4113473412049  
Prediction\_local [281.13312436]  
Right: 281.4789373771807  
Intercept 281.0021402742255  
Prediction\_local [281.94042494]  
Right: 281.8524655334203  
Intercept 281.32896025319883  
Prediction\_local [281.53650244]  
Right: 281.1479788437241  
Intercept 281.4097749837005  
Prediction\_local [281.48944408]  
Right: 281.5837661409229  
Intercept 281.13322544625544  
Prediction\_local [281.87572505]  
Right: 281.49431002378026  
Intercept 281.49760692258235  
Prediction\_local [281.25735683]  
Right: 281.23538311894174  
Intercept 281.3643431480532  
Prediction\_local [281.26874581]  
Right: 281.26373508270353  
Intercept 281.3470115426377  
Prediction\_local [281.61432928]  
Right: 281.28007590629  
Intercept 281.08260268840394  
Prediction\_local [281.69970166]  
Right: 281.90183750225356  
Intercept 281.5509922528838  
Prediction\_local [281.18116043]  
Right: 281.2341763480646  
Intercept 281.4920863251274  
Prediction\_local [280.83315608]  
Right: 281.0952675616551  
Intercept 281.46601559122143  
Prediction\_local [281.33329121]  
Right: 281.2338354898803  
Intercept 281.1606333768285  
Prediction\_local [281.76179193]  
Right: 281.791712792598  
Intercept 281.46664386121245  
Prediction\_local [281.13643783]  
Right: 281.32547121300956  
Intercept 281.231654360784  
Prediction\_local [281.80400198]  
Right: 281.8007330120915  
Intercept 281.473853366641  
Prediction\_local [281.1461668]  
Right: 281.07040634489016  
Intercept 281.2984877818897  
Prediction\_local [281.45843051]  
Right: 281.40595307003895  
Intercept 281.46122193749306  
Prediction\_local [281.24462108]  
Right: 281.0758652630242  
Intercept 281.34712113516366  
Prediction\_local [281.58695196]  
Right: 281.58010702736226  
Intercept 281.0853622995971  
Prediction\_local [281.90241381]  
Right: 282.05376217153616  
Intercept 281.01259403079246  
Prediction\_local [281.80116499]  
Right: 281.57454884308027  
Intercept 281.02624187379433  
Prediction\_local [281.74976973]  
Right: 281.80068265069156  
Intercept 281.4476233219411  
Prediction\_local [281.11489976]  
Right: 281.215571346475  
Intercept 281.1952901414617  
Prediction\_local [281.7649448]  
Right: 281.76571487704734  
Intercept 281.2550568464614  
Prediction\_local [281.67158464]  
Right: 281.7624500024097  
Intercept 281.31766005515055  
Prediction\_local [281.13725191]  
Right: 281.02346789541883  
Intercept 281.33575728784456  
Prediction\_local [281.2027732]  
Right: 281.37867933974064  
Intercept 281.22604632952545  
Prediction\_local [281.52998955]  
Right: 281.42221143893556  
Intercept 281.37880688879943  
Prediction\_local [281.7900819]  
Right: 281.6075876719041  
Intercept 281.2637799321514  
Prediction\_local [281.52409392]  
Right: 281.28958985739297  
Intercept 281.3086957192992  
Prediction\_local [281.38152172]  
Right: 281.4532356374221  
Intercept 281.64483339186734  
Prediction\_local [280.9855205]  
Right: 281.154085379992  
Intercept 281.49273034142726  
Prediction\_local [281.00647022]  
Right: 280.8716719583077  
Intercept 281.48626546048655  
Prediction\_local [281.50478787]  
Right: 281.5601667946449  
Intercept 281.31323837174205  
Prediction\_local [281.42741849]  
Right: 281.4429941219126  
Intercept 281.3799549458644  
Prediction\_local [281.55017205]  
Right: 281.51901369218143  
Intercept 281.42074391866385  
Prediction\_local [281.24186557]  
Right: 280.9550889496825  
Intercept 281.4980396118384  
Prediction\_local [281.12288762]  
Right: 281.1614906437269  
Intercept 281.3685310682446  
Prediction\_local [281.46869563]  
Right: 281.67027181751985  
Intercept 281.14386984707403  
Prediction\_local [282.13983596]  
Right: 281.99614791289974  
Intercept 281.493937152019  
Prediction\_local [281.02537081]  
Right: 280.9482204172489  
Intercept 281.35260674423677  
Prediction\_local [281.43052166]  
Right: 281.46846077980945  
Intercept 281.2194130497225  
Prediction\_local [281.8030665]  
Right: 281.5919173672913  
Intercept 281.4371019731871  
Prediction\_local [281.24299317]  
Right: 281.25550801517363  
Intercept 281.54163863340034  
Prediction\_local [281.14899948]  
Right: 281.1807653508943  
Intercept 281.3784049775187  
Prediction\_local [281.25690418]  
Right: 281.4410776079631  
Intercept 281.1626104796748  
Prediction\_local [281.55288476]  
Right: 281.1458647108846  
Intercept 281.3362613776049  
Prediction\_local [281.12278479]  
Right: 280.72630129487965  
Intercept 281.27350570611696  
Prediction\_local [281.30404482]  
Right: 281.22896309479336  
Intercept 281.7323255437363  
Prediction\_local [280.63901056]  
Right: 281.1045319326632  
Intercept 281.14640029883606  
Prediction\_local [281.86201828]  
Right: 281.9400580829764  
Intercept 281.24983300680935  
Prediction\_local [281.48218507]  
Right: 281.40913974146497  
Intercept 281.4195846804415  
Prediction\_local [281.30238728]  
Right: 281.593935208323  
Intercept 281.1235981552018  
Prediction\_local [281.71632208]  
Right: 281.8667270307526  
Intercept 281.6336760398335  
Prediction\_local [280.36123883]  
Right: 280.86854189653076  
Intercept 281.2268615696018  
Prediction\_local [281.71237688]  
Right: 281.17198725783925  
Intercept 281.67443229805656  
Prediction\_local [280.48709472]  
Right: 281.053423609802  
Intercept 281.49138417520817  
Prediction\_local [281.105049]  
Right: 281.1233339472918  
Intercept 281.3624588319254  
Prediction\_local [281.49173407]  
Right: 281.4695910949633  
Intercept 281.2483970097238  
Prediction\_local [281.30389888]  
Right: 281.14616008366306  
Intercept 281.43002440876995  
Prediction\_local [281.02926265]  
Right: 281.24663626797326  
Intercept 281.41239089051675  
Prediction\_local [281.0082717]  
Right: 281.17940479883544  
Intercept 281.25785672552905  
Prediction\_local [281.78811197]  
Right: 281.67325280866106  
Intercept 281.2296828181388  
Prediction\_local [281.69415442]  
Right: 281.6430708408698  
Intercept 281.38431318876167  
Prediction\_local [281.28981394]  
Right: 281.4204151549178  
Intercept 281.4231782736637  
Prediction\_local [281.44002786]  
Right: 281.36005527395616  
Intercept 281.0807228812845  
Prediction\_local [281.57553412]  
Right: 281.8861233921531  
Intercept 281.30263641615204  
Prediction\_local [281.7865291]  
Right: 281.7754863041675

exp

<lime.explanation.Explanation at 0x25d8c17cee0>

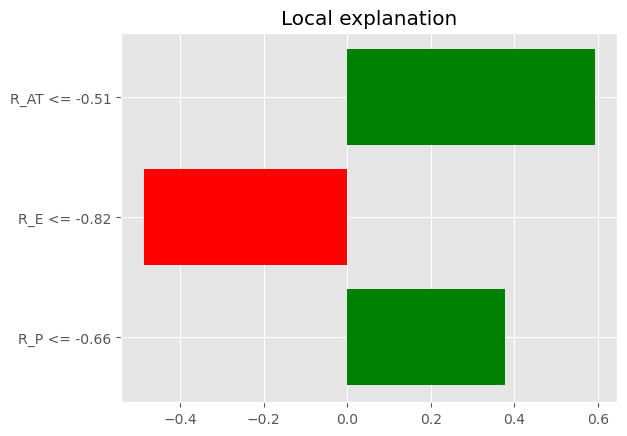
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

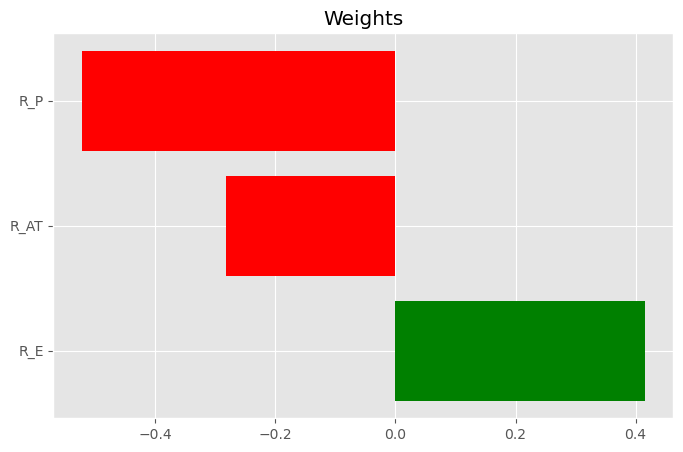
##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_R\_train.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_AT <= -0.51', 0.5930355490840262),  
 ('R\_E <= -0.82', -0.4866181327309232),  
 ('R\_P <= -0.66', 0.3774752639677357)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.5930355490840262),  
 (0, 0.4866181327309232),  
 (2, -0.3774752639677357)],  
 1: [(1, 0.5930355490840262),  
 (0, -0.4866181327309232),  
 (2, 0.3774752639677357)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.7865291]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.7754863041675

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_RR\_train.html')

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, lm1.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 281.4863772745298  
Prediction\_local [281.47424193]  
Right: 281.3943914439501  
Intercept 281.1108985361627  
Prediction\_local [281.54861082]  
Right: 281.5522921113873  
Intercept 281.5954387496208  
Prediction\_local [281.0160629]  
Right: 281.052272862199  
Intercept 281.0261659598412  
Prediction\_local [281.88757599]  
Right: 281.9265250578578  
Intercept 281.48565431866365  
Prediction\_local [281.17541169]  
Right: 281.09524173456464  
Intercept 280.91190490330746  
Prediction\_local [281.99432217]  
Right: 281.85472896849524  
Intercept 281.2504584021224  
Prediction\_local [282.00371014]  
Right: 281.6030199832242  
Intercept 281.63523043399454  
Prediction\_local [281.77975305]  
Right: 281.8226745019986  
Intercept 281.64101967168364  
Prediction\_local [281.28183816]  
Right: 280.81885025086115  
Intercept 281.18344293810577  
Prediction\_local [281.87443539]  
Right: 281.87946817157695  
Intercept 281.6705906672005  
Prediction\_local [281.11864898]  
Right: 281.0942010823298  
Intercept 281.308967593913  
Prediction\_local [281.41033937]  
Right: 281.7080784516426  
Intercept 281.2117861594253  
Prediction\_local [281.67023304]  
Right: 281.8231536161313  
Intercept 281.47666068771656  
Prediction\_local [281.12276788]  
Right: 281.45150859311275  
Intercept 281.5515083768719  
Prediction\_local [280.80308464]  
Right: 281.10499891676545  
Intercept 281.60869031452984  
Prediction\_local [281.43020152]  
Right: 281.1057159472339  
Intercept 281.02822688890166  
Prediction\_local [281.75350859]  
Right: 281.72877610854897  
Intercept 281.32506902517173  
Prediction\_local [281.91922702]  
Right: 281.84710216876476  
Intercept 281.3664404504643  
Prediction\_local [281.57165108]  
Right: 281.1186049129652  
Intercept 281.5069285109262  
Prediction\_local [281.00208865]  
Right: 280.7071891891553  
Intercept 281.07923542280315  
Prediction\_local [281.43348601]  
Right: 281.4353467035981  
Intercept 281.44882048134406  
Prediction\_local [281.10259908]  
Right: 281.17134001633195  
Intercept 281.1422601383289  
Prediction\_local [281.9373266]  
Right: 281.9295088955616  
Intercept 281.25233476346045  
Prediction\_local [281.90764443]  
Right: 281.8394235682465  
Intercept 281.1055857081959  
Prediction\_local [281.75038456]  
Right: 281.707339359358  
Intercept 281.55829582043464  
Prediction\_local [280.85933126]  
Right: 281.0405411013056  
Intercept 281.29157686275124  
Prediction\_local [280.95798605]  
Right: 280.6730722839087

exp

<lime.explanation.Explanation at 0x25d8d59b610>

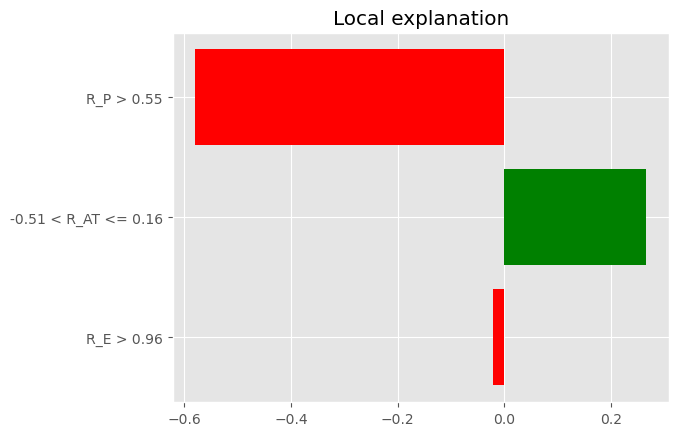
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

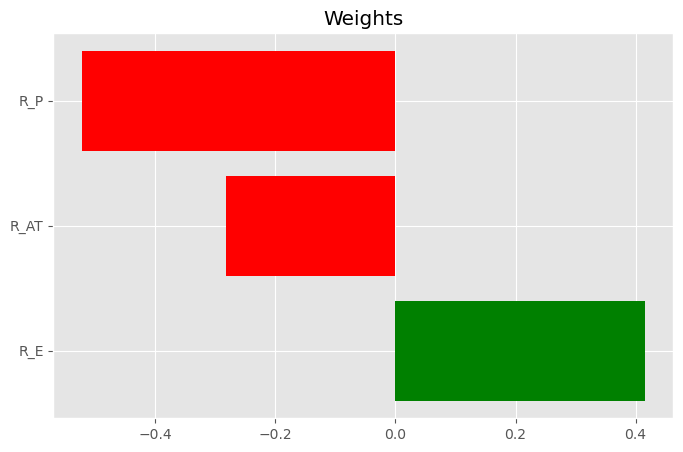
##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_R\_test.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P > 0.55', -0.5791842797363498),  
 ('-0.51 < R\_AT <= 0.16', 0.2665633334189921),  
 ('R\_E > 0.96', -0.020969862057348736)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 0.5791842797363498),  
 (1, -0.2665633334189921),  
 (0, 0.020969862057348736)],  
 1: [(2, -0.5791842797363498),  
 (1, 0.2665633334189921),  
 (0, -0.020969862057348736)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.95798605]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.6730722839087

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_RR\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

####### MODEL 2: GT Features (X\_G) and RS Target (LL\_R)#####################

# Instantiation using LinearRegression  
model = LinearRegression(fit\_intercept = True, copy\_X = True, positive = False, n\_jobs = None)

###### Training and fiting the linear model on the whole dataset  
lm2 = model.fit(X\_G, y\_R)

# The model interept  
print('The intercept is:', lm2.intercept\_)

The intercept is: 284.12634585428

# The model coefficients  
print('The model coefficients are:', lm2.coef\_)

The model coefficients are: [-0.0266713 -0.08304324 -0.00220106]

################ Finding metrics on the whole dataset

# The coefficient of determination  
  
print('The coefficient of determination is:', lm2.score(X\_G, y\_R))

The coefficient of determination is: 0.40443203373818526

# Making the prediction  
lm2\_pred = lm2.predict(X\_G)

# lm1 MAE  
print('The lm2 MAE is:%.2f'% mean\_absolute\_error(y\_R, lm2\_pred))

The lm2 MAE is:0.28

# lm1 MSE  
print('The lm2 MSE is:%.2f'% mean\_squared\_error(y\_R, lm2\_pred))

The lm2 MSE is:0.12

## The k-fold Cross-validation  
import numpy as np

# On the full dataset  
score\_lm2 = cross\_val\_score(lm2, X\_G, y\_R, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_lm2

array([-0.06670921, -0.10760063, -0.1232077 , -0.24679167, -0.13648838,  
 -0.10796731, -0.05510009, -0.2707046 , -0.11373093, -0.10907198])

# The absolute mean score on the full dataset

print('The CV MSE is:', absolute(np.mean(score\_lm2)))

The CV MSE is: 0.13373724757162614

# LIME

###### Initializing the explainer  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_G\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the full XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, lm2.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 281.23120224914936  
Prediction\_local [281.81669059]  
Right: 281.91499520368984  
Intercept 281.22869825221244  
Prediction\_local [281.78334414]  
Right: 281.75069668545353  
Intercept 281.37950994524886  
Prediction\_local [281.15240212]  
Right: 281.17469862145435  
Intercept 281.55819321380756  
Prediction\_local [281.1026773]  
Right: 281.0747437984792  
Intercept 281.44722145045273  
Prediction\_local [281.20172765]  
Right: 281.1630916668406  
Intercept 281.5183750970339  
Prediction\_local [281.33937352]  
Right: 281.26177401586125  
Intercept 281.502043351751  
Prediction\_local [281.22041797]  
Right: 281.26605480359774  
Intercept 281.49937445762293  
Prediction\_local [281.20681613]  
Right: 281.28110794125087  
Intercept 281.39595650307336  
Prediction\_local [281.62751132]  
Right: 281.4656066961886  
Intercept 281.28761590755823  
Prediction\_local [281.5087799]  
Right: 281.40240240171596  
Intercept 281.16474869492475  
Prediction\_local [281.78525896]  
Right: 281.7604267735956  
Intercept 281.2334641619641  
Prediction\_local [281.82485522]  
Right: 281.98467315963995  
Intercept 281.22500349129376  
Prediction\_local [281.83560238]  
Right: 281.9639419831263  
Intercept 281.23026872448776  
Prediction\_local [281.80698108]  
Right: 281.88001957609055  
Intercept 281.5371344817359  
Prediction\_local [281.15152035]  
Right: 281.19254863612025  
Intercept 281.5423581645763  
Prediction\_local [281.0239372]  
Right: 281.0003670819549  
Intercept 281.4940737401198  
Prediction\_local [281.08253517]  
Right: 281.07122713651665  
Intercept 281.4040794039101  
Prediction\_local [281.10977131]  
Right: 281.15775740615163  
Intercept 281.5709958510325  
Prediction\_local [281.0345261]  
Right: 281.1438351635186  
Intercept 281.4492343713812  
Prediction\_local [281.25628669]  
Right: 281.1503532968908  
Intercept 281.36547480404647  
Prediction\_local [281.42931805]  
Right: 281.4138231421613  
Intercept 281.2533845063088  
Prediction\_local [281.52747756]  
Right: 281.43768096243633  
Intercept 281.311708188175  
Prediction\_local [281.4837676]  
Right: 281.70023524041176  
Intercept 281.2290788453122  
Prediction\_local [281.85074051]  
Right: 282.0045548045356  
Intercept 281.2143070463802  
Prediction\_local [281.83463404]  
Right: 281.88268670643504  
Intercept 281.23050177010595  
Prediction\_local [281.75162032]  
Right: 281.6741994370817  
Intercept 281.502207425231  
Prediction\_local [281.00472826]  
Right: 281.1083232959635  
Intercept 281.47072456874173  
Prediction\_local [281.16306864]  
Right: 281.08456278823905  
Intercept 281.39092051498574  
Prediction\_local [281.15694361]  
Right: 281.16397083233124  
Intercept 281.5862691478636  
Prediction\_local [281.02473885]  
Right: 281.2749404708887  
Intercept 281.4000705579254  
Prediction\_local [281.40675042]  
Right: 281.41819777894324  
Intercept 281.3790432553263  
Prediction\_local [281.44645228]  
Right: 281.24812826790384  
Intercept 281.4387997630622  
Prediction\_local [281.45029943]  
Right: 281.4086064431586  
Intercept 281.2323814349603  
Prediction\_local [281.62457604]  
Right: 281.382853323814  
Intercept 281.37137511223585  
Prediction\_local [281.49281506]  
Right: 281.4739868580103  
Intercept 281.2601385586977  
Prediction\_local [281.77270765]  
Right: 281.9535763948694  
Intercept 281.2485271677523  
Prediction\_local [281.75664091]  
Right: 281.9900666880741  
Intercept 281.2173790326743  
Prediction\_local [281.78870129]  
Right: 282.00846330123693  
Intercept 281.4055117299527  
Prediction\_local [281.17192064]  
Right: 281.14244939194475  
Intercept 281.4375501171883  
Prediction\_local [281.09670456]  
Right: 281.032343012216  
Intercept 281.6125209140382  
Prediction\_local [281.21710459]  
Right: 281.1453204896612  
Intercept 281.5403972317854  
Prediction\_local [281.12495607]  
Right: 281.2880480372187  
Intercept 281.5527833811829  
Prediction\_local [281.2587324]  
Right: 281.31726500420183  
Intercept 281.3985042034596  
Prediction\_local [281.51257103]  
Right: 281.2090055787966  
Intercept 281.3559329754212  
Prediction\_local [281.64603779]  
Right: 281.5618338255174  
Intercept 281.2699348145925  
Prediction\_local [281.36710963]  
Right: 281.42728574030684  
Intercept 281.36257750050885  
Prediction\_local [281.53680458]  
Right: 281.54742514129873  
Intercept 281.2767046744054  
Prediction\_local [281.84815953]  
Right: 281.7293300088251  
Intercept 281.2452253096783  
Prediction\_local [281.78557317]  
Right: 281.73381473789607  
Intercept 281.2087138973776  
Prediction\_local [281.74959494]  
Right: 281.64961903161185  
Intercept 281.43864277396494  
Prediction\_local [281.22216958]  
Right: 281.2892600963814  
Intercept 281.52757570577876  
Prediction\_local [281.01298034]  
Right: 280.989698560577  
Intercept 281.50484174369717  
Prediction\_local [281.1486256]  
Right: 281.2528949330622  
Intercept 281.4550438678466  
Prediction\_local [281.42616255]  
Right: 281.3443995789784  
Intercept 281.4976302141892  
Prediction\_local [281.32409243]  
Right: 281.2395450963801  
Intercept 281.4880676291408  
Prediction\_local [281.15469007]  
Right: 281.2638091257304  
Intercept 281.4438769947466  
Prediction\_local [281.42434783]  
Right: 281.3954265289984  
Intercept 281.28737693612356  
Prediction\_local [281.46545334]  
Right: 281.39615934166375  
Intercept 281.32391201225136  
Prediction\_local [281.57470107]  
Right: 281.66520034506726  
Intercept 281.20359620315287  
Prediction\_local [281.86513219]  
Right: 281.84765181109054  
Intercept 281.21835055791996  
Prediction\_local [281.78290714]  
Right: 281.70153587451387  
Intercept 281.3333509467449  
Prediction\_local [281.35943945]  
Right: 281.5149915141585  
Intercept 281.46166485963147  
Prediction\_local [281.19951335]  
Right: 281.124658645024  
Intercept 281.49897762885234  
Prediction\_local [281.05609034]  
Right: 280.97542447561875  
Intercept 281.40937728389326  
Prediction\_local [281.27402567]  
Right: 281.24131761232104  
Intercept 281.537310155535  
Prediction\_local [281.12895146]  
Right: 281.19437860920544  
Intercept 281.4967065183325  
Prediction\_local [281.16431406]  
Right: 281.2842817040105  
Intercept 281.3817869267096  
Prediction\_local [281.37194674]  
Right: 281.1649089609597  
Intercept 281.3959385589368  
Prediction\_local [281.4868386]  
Right: 281.51888900915947  
Intercept 281.37669912046414  
Prediction\_local [281.5567931]  
Right: 281.8583569559182  
Intercept 281.3760840272422  
Prediction\_local [281.36420864]  
Right: 281.59406699125697  
Intercept 281.20147255222514  
Prediction\_local [281.80016783]  
Right: 281.7331792377813  
Intercept 281.19535184385234  
Prediction\_local [281.79490184]  
Right: 281.7077493006934  
Intercept 281.2869241184395  
Prediction\_local [281.77270004]  
Right: 281.6084003439604  
Intercept 281.4077957912559  
Prediction\_local [281.16151001]  
Right: 281.1554821105461  
Intercept 281.59543296128976  
Prediction\_local [281.02879784]  
Right: 281.0030342122994  
Intercept 281.4924486682806  
Prediction\_local [281.24719086]  
Right: 281.1064464294918  
Intercept 281.5633257204562  
Prediction\_local [281.03719513]  
Right: 281.15531968771967  
Intercept 281.5177098165718  
Prediction\_local [281.28969816]  
Right: 281.35832409560385  
Intercept 281.48379681062875  
Prediction\_local [281.23776461]  
Right: 281.1314584783799  
Intercept 281.4559260116186  
Prediction\_local [281.50680975]  
Right: 281.47834659816954  
Intercept 281.4830433695706  
Prediction\_local [281.35096224]  
Right: 281.37812655623384  
Intercept 281.3658955323888  
Prediction\_local [281.41847982]  
Right: 281.6219203933135  
Intercept 281.20277704429884  
Prediction\_local [281.81642939]  
Right: 281.71835855432613  
Intercept 281.26193041973875  
Prediction\_local [281.74540963]  
Right: 281.8130383843575  
Intercept 281.1766791750894  
Prediction\_local [281.77527943]  
Right: 281.826676969201  
Intercept 281.4116781877657  
Prediction\_local [281.1586407]  
Right: 281.20085296027474  
Intercept 281.44250778743435  
Prediction\_local [281.08022802]  
Right: 280.94669190807167  
Intercept 281.4716325696286  
Prediction\_local [281.15410738]  
Right: 281.1954001640954  
Intercept 281.53687952046783  
Prediction\_local [281.23185511]  
Right: 281.2732645196003  
Intercept 281.4835109655635  
Prediction\_local [281.31631733]  
Right: 281.35191561290594  
Intercept 281.437626070586  
Prediction\_local [281.22351654]  
Right: 281.25375052885613  
Intercept 281.47431869789693  
Prediction\_local [281.37300726]  
Right: 281.46416511017713  
Intercept 281.17039818656707  
Prediction\_local [281.69563875]  
Right: 281.56870291630935  
Intercept 281.323840331661  
Prediction\_local [281.58324228]  
Right: 281.6337710133031  
Intercept 281.2313509920407  
Prediction\_local [281.83056638]  
Right: 281.9209056967483  
Intercept 281.26688796045215  
Prediction\_local [281.7278588]  
Right: 281.8317082967689  
Intercept 281.2427762569858  
Prediction\_local [281.73545295]  
Right: 281.77742725664353  
Intercept 281.39005079771994  
Prediction\_local [281.28587447]  
Right: 281.15872547326006  
Intercept 281.4532464580967  
Prediction\_local [281.08808963]  
Right: 281.0178548957545  
Intercept 281.3652536107004  
Prediction\_local [281.14072653]  
Right: 281.16281836759214  
Intercept 281.456522303884  
Prediction\_local [281.46717194]  
Right: 281.32554273676925  
Intercept 281.5369754712273  
Prediction\_local [281.2107096]  
Right: 281.22169789809357  
Intercept 281.39931821507724  
Prediction\_local [281.25456202]  
Right: 281.35823247837897  
Intercept 281.49047000661545  
Prediction\_local [281.37816988]  
Right: 281.2641270602129  
Intercept 281.3478240117384  
Prediction\_local [281.59421627]  
Right: 281.48035504794797  
Intercept 281.3123686351978  
Prediction\_local [281.6078178]  
Right: 281.6094342732091  
Intercept 281.17591383700295  
Prediction\_local [281.90313241]  
Right: 281.8770198787524

exp

<lime.explanation.Explanation at 0x1d581b73eb0>

### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()

# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_G.columns);  
 plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= 25.38', 0.5950597225024261),  
 ('G\_P <= 0.00', 0.16328015020867231),  
 ('9.70 < G\_E <= 13.62', -0.10594770370687108)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.5950597225024261),  
 (2, -0.16328015020867231),  
 (0, 0.10594770370687108)],  
 1: [(1, 0.5950597225024261),  
 (2, 0.16328015020867231),  
 (0, -0.10594770370687108)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.85348672]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.8770198787524

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_GR\_Whole.html')

###########################################################################################################

###### Training and fiting the linear model on the training dataset

# Instantiation using LinearRegression  
model = LinearRegression(fit\_intercept = True, copy\_X = True, positive = False, n\_jobs = None)#, normalize = False)

###### Training and fiting the linear model on the training dataset  
lm2 = model.fit(X\_G\_train, y\_R\_train)

#### Predicting on the training dataset  
lm2\_predtr = lm2.predict(X\_G\_train)

# The training data metrics

# The R-sq on the trainong dataset  
  
print('The training R2 is:', lm2.score(X\_G\_train, y\_R\_train))

The training R2 is: 0.4207632638784464

# The training MAE  
mae\_tr = mean\_absolute\_error(y\_G\_train, lm2\_predtr)

print('The training MAE is:', mae\_tr)

The training MAE is: 1.6560493827160394

# The training MSE  
mse\_tr = mean\_squared\_error(y\_G\_train, lm2\_predtr)

print('The training MSE is:', mse\_tr)

The training MSE is: 2.9192174391308146

####### Predicting on the testing dataset  
  
lm2\_predts = lm2.predict(X\_G\_test)

# The testing data metrics

# The R-sq on the testing dataset  
print('The testing R2 is:', lm2.score(X\_G\_test, y\_R\_test))

The testing R2 is: 0.05939233531593413

# The testing MAE  
mae\_ts = mean\_absolute\_error(y\_R\_test, lm2\_predts)

print('The testing MAE is:', mae\_ts)

The testing MAE is: 0.2832669734392749

##### The Testing MSE  
mse\_ts = mean\_squared\_error(y\_R\_test, lm2\_predts)

#### Printing the MSE  
print(' The Testing MSE is:', mse\_ts)

The Testing MSE is: 0.1366180376972227

## The k-fold Cross-validation

# On the training dataset  
score\_train = cross\_val\_score(lm2, X\_G\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_train

array([-0.09247887, -0.08414828, -0.17319182, -0.12717762, -0.15150766,  
 -0.14686387, -0.08567474, -0.04310685, -0.22247188, -0.26403897])

# The absolute mean score on the training dataset

print('The training CV MSE is:', absolute(np.mean(score\_train)))

The training CV MSE is: 0.13906605548165857

## On the testing dataset  
score\_test = cross\_val\_score(lm2, X\_G\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

print('The testing CV MSE is:', absolute(np.mean(score\_test)))

The testing CV MSE is: 0.10866027433987543

# LIME

# 1. Creating Explainer object

###### Initializing the explainer  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_R\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the full training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, lm2.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.26976668005176  
Prediction\_local [281.0885737]  
Right: 281.1124082388573  
Intercept 281.35690758167397  
Prediction\_local [280.94487175]  
Right: 280.9761653382433  
Intercept 281.1355762833654  
Prediction\_local [281.74805406]  
Right: 281.81673188216706  
Intercept 281.3798717486112  
Prediction\_local [281.10404493]  
Right: 281.24229932419235  
Intercept 281.3734665068887  
Prediction\_local [281.39950505]  
Right: 281.0945830400884  
Intercept 281.1182787800667  
Prediction\_local [281.74189623]  
Right: 281.7338542439663  
Intercept 281.26632032968473  
Prediction\_local [281.44164591]  
Right: 281.278744584984  
Intercept 281.44819642498675  
Prediction\_local [281.17076132]  
Right: 281.13638806973694  
Intercept 281.3455271427071  
Prediction\_local [281.11967816]  
Right: 281.2364887267872  
Intercept 281.3585521231351  
Prediction\_local [281.22878994]  
Right: 281.3124837285764  
Intercept 281.19832706885194  
Prediction\_local [281.59652947]  
Right: 281.45723176569714  
Intercept 281.36664029329614  
Prediction\_local [280.99387629]  
Right: 280.9365948446772  
Intercept 281.34479696113425  
Prediction\_local [281.08165573]  
Right: 281.1590166892405  
Intercept 281.29176957991876  
Prediction\_local [281.26977757]  
Right: 281.2835348140393  
Intercept 281.20488942082267  
Prediction\_local [281.6808679]  
Right: 281.7003513286026  
Intercept 281.3969926599639  
Prediction\_local [280.8667303]  
Right: 281.09739237703707  
Intercept 281.2867513213669  
Prediction\_local [281.33492827]  
Right: 281.43651792121824  
Intercept 281.28083229202923  
Prediction\_local [281.51089828]  
Right: 281.4863269501527  
Intercept 281.4671247634309  
Prediction\_local [281.25388046]  
Right: 281.23155842086277  
Intercept 281.2738434812804  
Prediction\_local [281.3737216]  
Right: 281.35559404645846  
Intercept 281.41216994478924  
Prediction\_local [281.08943544]  
Right: 280.98431269215587  
Intercept 281.2093244477357  
Prediction\_local [281.7401156]  
Right: 281.6840986883678  
Intercept 281.45379794601706  
Prediction\_local [280.89124512]  
Right: 281.150672696463  
Intercept 281.4343861626203  
Prediction\_local [280.8841699]  
Right: 280.94481189320084  
Intercept 281.3193858452512  
Prediction\_local [281.10125942]  
Right: 281.065364443154  
Intercept 281.16315293482967  
Prediction\_local [281.59021423]  
Right: 281.7891423444726  
Intercept 281.33601685267627  
Prediction\_local [281.11803723]  
Right: 281.2425303492104  
Intercept 281.1408698941338  
Prediction\_local [281.64770009]  
Right: 281.6941840624258  
Intercept 281.3369179537218  
Prediction\_local [281.19601577]  
Right: 281.14751909237685  
Intercept 281.17628476244107  
Prediction\_local [281.64411119]  
Right: 281.63462427755627  
Intercept 281.38505756763396  
Prediction\_local [281.16270572]  
Right: 281.1231656295793  
Intercept 281.0952359802214  
Prediction\_local [281.72344232]  
Right: 281.67099315334826  
Intercept 281.17649187342363  
Prediction\_local [281.63872001]  
Right: 281.950979512993  
Intercept 281.1025172826672  
Prediction\_local [281.45363537]  
Right: 281.6089754970912  
Intercept 281.1270772147113  
Prediction\_local [281.67506263]  
Right: 281.6954628421068  
Intercept 281.27611554364114  
Prediction\_local [281.47400937]  
Right: 281.3785279180946  
Intercept 281.12961956020064  
Prediction\_local [281.6252528]  
Right: 281.5617050691597  
Intercept 281.14200945882743  
Prediction\_local [281.49713906]  
Right: 281.381946310819  
Intercept 281.3039602289236  
Prediction\_local [281.06878703]  
Right: 280.9978687545478  
Intercept 281.1499162244414  
Prediction\_local [281.83953477]  
Right: 281.84199730710003  
Intercept 281.30746444138106  
Prediction\_local [281.40987138]  
Right: 281.58724482450725  
Intercept 281.32836857942857  
Prediction\_local [281.52814514]  
Right: 281.5375592826513  
Intercept 281.19619069070956  
Prediction\_local [281.42048718]  
Right: 281.3780554592172  
Intercept 281.32657052488594  
Prediction\_local [281.47434405]  
Right: 281.64191612288096  
Intercept 281.2496968508365  
Prediction\_local [281.13285049]  
Right: 281.13897288537834  
Intercept 281.32527149639213  
Prediction\_local [281.2095103]  
Right: 281.16385475014584  
Intercept 281.25535167800047  
Prediction\_local [281.51160845]  
Right: 281.35822036844695  
Intercept 281.2263584826165  
Prediction\_local [281.3293339]  
Right: 281.2437626529756  
Intercept 281.1490938043065  
Prediction\_local [281.63397503]  
Right: 281.84278875167286  
Intercept 281.42461274338854  
Prediction\_local [281.1194628]  
Right: 281.0687883558005  
Intercept 281.4226989514004  
Prediction\_local [280.89961832]  
Right: 280.95689253704694  
Intercept 281.0105491044818  
Prediction\_local [281.78796342]  
Right: 281.9544077629953  
Intercept 281.20445545045214  
Prediction\_local [281.67917254]  
Right: 281.768001217742  
Intercept 281.2480257841115  
Prediction\_local [281.10729554]  
Right: 281.2265575532624  
Intercept 281.2404872676598  
Prediction\_local [281.42102646]  
Right: 281.3538110287971  
Intercept 281.37803183571987  
Prediction\_local [281.13724058]  
Right: 281.1187297055673  
Intercept 281.32839985480393  
Prediction\_local [280.98754946]  
Right: 281.04693507631856  
Intercept 281.42541750668005  
Prediction\_local [280.98309245]  
Right: 281.0780346328534  
Intercept 281.35554997805303  
Prediction\_local [281.24871977]  
Right: 281.14772192913097  
Intercept 281.4286379078114  
Prediction\_local [280.98737557]  
Right: 281.031834271511  
Intercept 281.35866191095266  
Prediction\_local [281.13090108]  
Right: 281.0791319557219  
Intercept 281.4182742285456  
Prediction\_local [281.16467224]  
Right: 281.2504229499705  
Intercept 281.3958111242875  
Prediction\_local [280.97886739]  
Right: 281.0379263267846  
Intercept 281.13840487427836  
Prediction\_local [281.64175069]  
Right: 281.9160163611358  
Intercept 281.22391437452825  
Prediction\_local [281.28231835]  
Right: 281.4654680074318  
Intercept 281.35652645094484  
Prediction\_local [281.30068872]  
Right: 281.1096173689784  
Intercept 281.23111633860196  
Prediction\_local [281.67597718]  
Right: 281.6041415231541  
Intercept 281.37868426895386  
Prediction\_local [281.03593769]  
Right: 281.09575786419896  
Intercept 281.4539935378839  
Prediction\_local [281.17736536]  
Right: 281.1836653674749  
Intercept 281.37103095079146  
Prediction\_local [281.22129881]  
Right: 281.2401543779046  
Intercept 281.34150118521836  
Prediction\_local [281.02491553]  
Right: 281.04217353407654  
Intercept 281.2046433015696  
Prediction\_local [281.52608919]  
Right: 281.5286826745318  
Intercept 281.44957231078706  
Prediction\_local [280.94355779]  
Right: 281.1303477478139  
Intercept 281.331817629092  
Prediction\_local [281.30377585]  
Right: 281.3030417361733  
Intercept 281.4542853511712  
Prediction\_local [281.08112104]  
Right: 281.1415849572991  
Intercept 281.1240140265641  
Prediction\_local [281.7459315]  
Right: 281.97268292929755  
Intercept 281.19333324500155  
Prediction\_local [281.76952835]  
Right: 281.92881349508895  
Intercept 281.3119414451614  
Prediction\_local [281.43842867]  
Right: 281.4680979984889  
Intercept 281.25905242545724  
Prediction\_local [281.42588827]  
Right: 281.41171257511405  
Intercept 281.12658494673497  
Prediction\_local [281.82655449]  
Right: 281.8452193667811  
Intercept 281.13143275536834  
Prediction\_local [281.67244814]  
Right: 281.9681028438681

exp

<lime.explanation.Explanation at 0x25d8d626490>

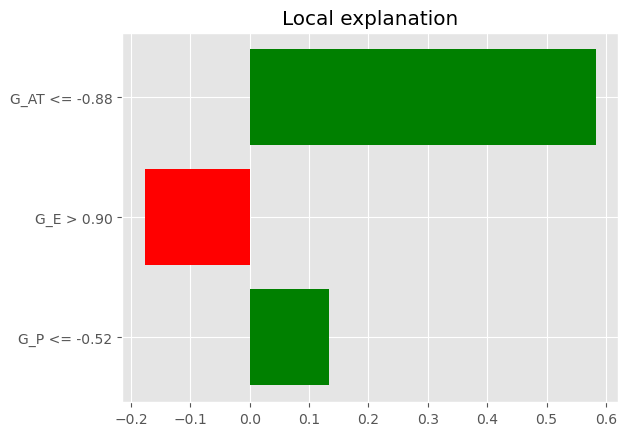
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

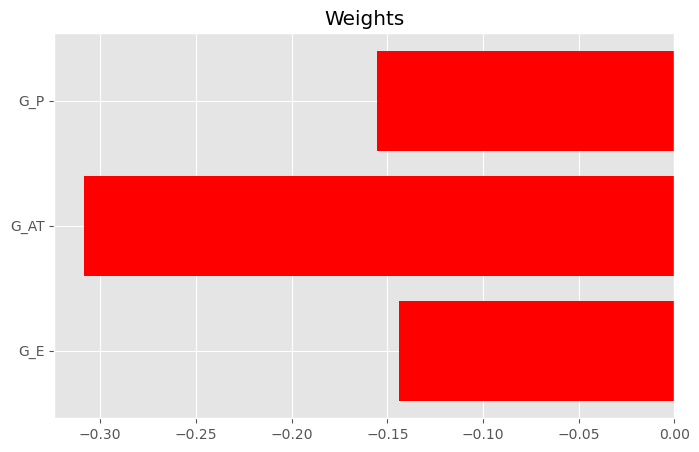
##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_G\_train.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= -0.88', 0.5831780277085343),  
 ('G\_E > 0.90', -0.17634355022983034),  
 ('G\_P <= -0.52', 0.13418090257619644)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.5831780277085343),  
 (0, 0.17634355022983034),  
 (2, -0.13418090257619644)],  
 1: [(1, 0.5831780277085343),  
 (0, -0.17634355022983034),  
 (2, 0.13418090257619644)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.67244814]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.9681028438681

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_GR\_train.html')

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, lm2.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 281.1732785779493  
Prediction\_local [281.56494741]  
Right: 281.71488984383626  
Intercept 281.1433008403138  
Prediction\_local [281.7167551]  
Right: 281.88133435406553  
Intercept 281.39069145581965  
Prediction\_local [281.14634365]  
Right: 280.90194009388625  
Intercept 281.33587187590217  
Prediction\_local [281.4192736]  
Right: 281.4471410596048  
Intercept 281.4349927634558  
Prediction\_local [281.27891732]  
Right: 281.30500361085683  
Intercept 281.3521198734233  
Prediction\_local [281.38176582]  
Right: 281.39565124896967  
Intercept 281.3847107966268  
Prediction\_local [281.12445224]  
Right: 281.3436129605531  
Intercept 281.12356324463144  
Prediction\_local [281.61824697]  
Right: 281.6625494725964  
Intercept 281.3016500316963  
Prediction\_local [281.2616832]  
Right: 281.01164982489064  
Intercept 281.1541495953605  
Prediction\_local [281.64749753]  
Right: 281.84580891263437  
Intercept 281.21979143513386  
Prediction\_local [281.0790055]  
Right: 281.13237027388135  
Intercept 281.03298129323747  
Prediction\_local [281.59271942]  
Right: 281.56231253414927  
Intercept 281.2062150007866  
Prediction\_local [281.64167262]  
Right: 281.7356228815261  
Intercept 281.35693451251757  
Prediction\_local [281.2845342]  
Right: 281.5950265277108  
Intercept 281.33697923575596  
Prediction\_local [281.05491776]  
Right: 281.09659094394266  
Intercept 281.30626660964697  
Prediction\_local [281.12624421]  
Right: 281.1719135111681  
Intercept 281.20405417805944  
Prediction\_local [281.16380038]  
Right: 281.2471810505399  
Intercept 281.44143356569253  
Prediction\_local [281.47923927]  
Right: 281.41977542310457  
Intercept 281.2680010351673  
Prediction\_local [281.25501366]  
Right: 280.9359296496454  
Intercept 281.3829537744881  
Prediction\_local [281.03026495]  
Right: 281.1182462249955  
Intercept 281.30252784836307  
Prediction\_local [281.34489873]  
Right: 281.5629565925613  
Intercept 281.3407350451813  
Prediction\_local [281.28912476]  
Right: 281.1990362893251  
Intercept 281.19132747661473  
Prediction\_local [281.72817646]  
Right: 281.65392433503195  
Intercept 281.17079260895275  
Prediction\_local [281.57276885]  
Right: 281.7823855324425  
Intercept 281.3681666391803  
Prediction\_local [281.68140692]  
Right: 281.89067598173705  
Intercept 281.19880194886076  
Prediction\_local [280.97155674]  
Right: 280.95991269800845  
Intercept 281.3427936481383  
Prediction\_local [281.28950637]  
Right: 280.9355948521046

exp

<lime.explanation.Explanation at 0x25d8d67e760>

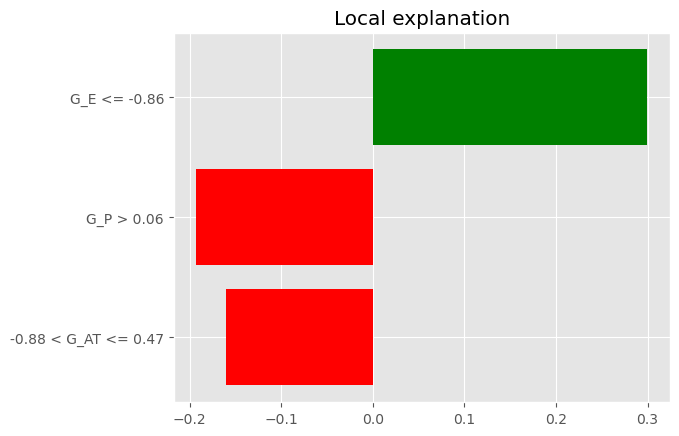
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

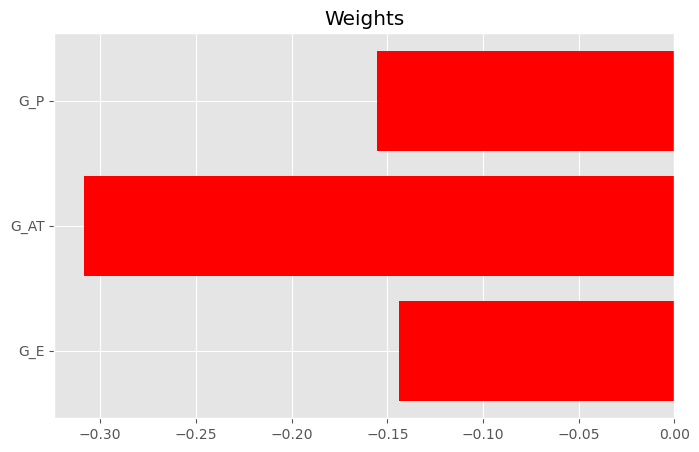
##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_G\_test.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_E <= -0.86', 0.2995980335101239),  
 ('G\_P > 0.06', -0.19263554645594053),  
 ('-0.88 < G\_AT <= 0.47', -0.16024976917258296)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(0, -0.2995980335101239),  
 (2, 0.19263554645594053),  
 (1, 0.16024976917258296)],  
 1: [(0, 0.2995980335101239),  
 (2, -0.19263554645594053),  
 (1, -0.16024976917258296)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.28950637]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.9355948521046

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_GR\_test.html')

###################################################################################################################

###################################################################################################################

###################################################################################################################

##################### MODEL 3: RS Features (X\_R) and GT Target (LL\_G)#####################

###### Training and fitting the linear model on the whole dataset

# Instantiation using LinearRegression  
model = LinearRegression(fit\_intercept = True, copy\_X = True, positive = False, n\_jobs = None)

###### Training and fiting the linear model on the whole dataset  
lm3 = model.fit(X\_R, y\_G)

# The model interept  
print('The intercept is:', lm3.intercept\_)

The intercept is: 282.4651919531141

# The model coefficients  
print('The model coefficients are:', lm3.coef\_)

The model coefficients are: [ 0.00479478 -0.08969274 -0.00619966]

################ Finding metrics on the whole dataset

# The coefficient of determination  
  
print('The coefficient of determination is:', lm3.score(X\_R, y\_G))

The coefficient of determination is: 0.5159856364020856

# Making the prediction  
lm3\_pred = lm3.predict(X\_R)

# lm1 MAE  
print('The lm3 MAE is:%.2f'% mean\_absolute\_error(y\_G, lm3\_pred))

The lm3 MAE is:0.31

# lm1 MSE  
print('The lm3 MSE is:%.2f'% mean\_squared\_error(y\_G, lm3\_pred))

The lm3 MSE is:0.15

## The k-fold Cross-validation  
import numpy as np

# On the full dataset  
score\_lm3 = cross\_val\_score(lm3, X\_R, y\_G, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_lm3

array([-0.11786545, -0.10129886, -0.29744168, -0.22205269, -0.11894325,  
 -0.04672674, -0.15986767, -0.21715851, -0.35591276, -0.08700057])

# The absolute mean score on the full dataset

print('The CV MSE is:', absolute(np.mean(score\_lm3)))

The CV MSE is: 0.17242681813618965

# LIME

###### Initializing the explainer  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_R\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the full XG datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, lm3.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 279.6953380713266  
Prediction\_local [280.10345272]  
Right: 280.074505858012  
Intercept 279.78078879309425  
Prediction\_local [279.83643305]  
Right: 279.93656492002395  
Intercept 279.7171939723577  
Prediction\_local [279.91243406]  
Right: 279.74523496117456  
Intercept 279.78946828789816  
Prediction\_local [279.58743901]  
Right: 279.68341844333344  
Intercept 279.7690438235827  
Prediction\_local [279.66102964]  
Right: 279.7047760668026  
Intercept 279.767071636881  
Prediction\_local [279.8018782]  
Right: 279.6993169449864  
Intercept 279.9063542599326  
Prediction\_local [279.47430756]  
Right: 279.54972333817847  
Intercept 279.8080368292718  
Prediction\_local [279.73666517]  
Right: 279.34557879332795  
Intercept 279.5970595204758  
Prediction\_local [280.09062906]  
Right: 279.954142609759  
Intercept 279.5970954843187  
Prediction\_local [280.15946581]  
Right: 280.07773580983473  
Intercept 279.66244945467304  
Prediction\_local [279.89390424]  
Right: 280.0377580927344  
Intercept 279.68948263516097  
Prediction\_local [280.13212615]  
Right: 280.17724062240336  
Intercept 279.61500687388997  
Prediction\_local [280.33338428]  
Right: 280.157692236767  
Intercept 279.686975729287  
Prediction\_local [279.98762607]  
Right: 280.04802728864865  
Intercept 279.71184461854676  
Prediction\_local [279.82526974]  
Right: 279.79475497580535  
Intercept 279.8699954050414  
Prediction\_local [279.58866662]  
Right: 279.6986471456336  
Intercept 279.7812376437709  
Prediction\_local [279.5223251]  
Right: 279.6445590547682  
Intercept 279.74406040134534  
Prediction\_local [279.79243262]  
Right: 279.65468254202784  
Intercept 279.9451728477692  
Prediction\_local [279.26531748]  
Right: 279.43561562122875  
Intercept 279.8083473451172  
Prediction\_local [279.66696188]  
Right: 279.4024939956234  
Intercept 279.60679266435585  
Prediction\_local [280.17476702]  
Right: 279.9547310257874  
Intercept 279.77603433025683  
Prediction\_local [279.82164089]  
Right: 280.02954154790876  
Intercept 279.626966871908  
Prediction\_local [280.05451646]  
Right: 280.06688526901615  
Intercept 279.59883546234295  
Prediction\_local [280.14958341]  
Right: 280.1787569670322  
Intercept 279.61261945303636  
Prediction\_local [280.26565341]  
Right: 280.36696162013175  
Intercept 279.7441410457055  
Prediction\_local [279.91571235]  
Right: 279.9479018114571  
Intercept 279.6598161252149  
Prediction\_local [279.86242971]  
Right: 279.7954662126913  
Intercept 279.91658832071147  
Prediction\_local [279.73998656]  
Right: 279.8131987709664  
Intercept 279.8475224265548  
Prediction\_local [279.4551431]  
Right: 279.68870546682274  
Intercept 279.6897742114307  
Prediction\_local [279.74060175]  
Right: 279.6241833730167  
Intercept 279.92662072359036  
Prediction\_local [279.33626064]  
Right: 279.4186123703485  
Intercept 279.8994252236648  
Prediction\_local [279.30736939]  
Right: 279.0718564318951  
Intercept 279.8278913993108  
Prediction\_local [279.43382428]  
Right: 279.70193654809935  
Intercept 279.7657814616361  
Prediction\_local [279.71715357]  
Right: 279.8608352333372  
Intercept 279.5191548106336  
Prediction\_local [280.27249123]  
Right: 280.0802581318758  
Intercept 279.6062158671509  
Prediction\_local [280.28440987]  
Right: 280.41593967557714  
Intercept 279.5018451210446  
Prediction\_local [280.31355322]  
Right: 280.5196767653141  
Intercept 279.6897161212448  
Prediction\_local [280.09904892]  
Right: 280.27453645377983  
Intercept 279.9196528190521  
Prediction\_local [279.60266772]  
Right: 279.6802894792564  
Intercept 279.84972828694316  
Prediction\_local [279.73858266]  
Right: 279.56670983500214  
Intercept 279.91133887619924  
Prediction\_local [279.38249426]  
Right: 279.4858717626177  
Intercept 280.01086265217464  
Prediction\_local [278.86601482]  
Right: 279.27500253790606  
Intercept 279.97796166910837  
Prediction\_local [279.21136736]  
Right: 279.16241711370697  
Intercept 279.8988784421332  
Prediction\_local [279.37617097]  
Right: 279.0001015914804  
Intercept 279.7960415667765  
Prediction\_local [279.77553254]  
Right: 279.63350352482604  
Intercept 279.7255159660033  
Prediction\_local [279.75010422]  
Right: 279.7537499271356  
Intercept 279.619364077404  
Prediction\_local [279.95038437]  
Right: 279.983056179499  
Intercept 279.57010591335734  
Prediction\_local [280.20595374]  
Right: 280.2927614974451  
Intercept 279.5647238832442  
Prediction\_local [280.18264961]  
Right: 280.2973131035164  
Intercept 279.54698457754284  
Prediction\_local [280.20648323]  
Right: 280.3553942118168  
Intercept 279.632063892739  
Prediction\_local [279.95055445]  
Right: 279.98107899401157  
Intercept 279.804351338179  
Prediction\_local [279.50002102]  
Right: 279.6624280092108  
Intercept 279.9175766332855  
Prediction\_local [279.53610953]  
Right: 279.5724188585272  
Intercept 279.9582034029404  
Prediction\_local [279.08610865]  
Right: 279.3723035956107  
Intercept 279.9725422409291  
Prediction\_local [279.03304519]  
Right: 279.1441528534052  
Intercept 279.8888862174185  
Prediction\_local [279.21084985]  
Right: 279.0697402425033  
Intercept 280.03071502391794  
Prediction\_local [278.96455711]  
Right: 279.4102270921689  
Intercept 279.8802409844399  
Prediction\_local [279.54069686]  
Right: 279.7406197797807  
Intercept 279.7467828895143  
Prediction\_local [279.8506445]  
Right: 279.9683620444223  
Intercept 279.53801358552266  
Prediction\_local [280.23256738]  
Right: 280.163916744134  
Intercept 279.6589695002494  
Prediction\_local [280.25952729]  
Right: 280.4124241282805  
Intercept 279.648338359969  
Prediction\_local [279.87823694]  
Right: 279.9511159910625  
Intercept 279.8419655133792  
Prediction\_local [279.6052738]  
Right: 279.75111594094307  
Intercept 279.7872231478538  
Prediction\_local [279.68955007]  
Right: 279.60449003980096  
Intercept 279.7338965932577  
Prediction\_local [279.46511045]  
Right: 279.43807392530243  
Intercept 280.0569603465699  
Prediction\_local [278.84775626]  
Right: 279.20683419019144  
Intercept 279.93214402591764  
Prediction\_local [279.22312064]  
Right: 279.15401388397686  
Intercept 279.88466839164124  
Prediction\_local [279.26901338]  
Right: 278.8443467301634  
Intercept 279.6690840609151  
Prediction\_local [279.68255902]  
Right: 279.59462739422116  
Intercept 279.892400411772  
Prediction\_local [279.58666614]  
Right: 279.7208732623059  
Intercept 279.75713667237454  
Prediction\_local [279.97629474]  
Right: 279.9511255796975  
Intercept 279.60443384786026  
Prediction\_local [280.16583921]  
Right: 280.33889020195437  
Intercept 279.59173293201974  
Prediction\_local [280.27396752]  
Right: 280.31751671655826  
Intercept 279.566307122899  
Prediction\_local [280.25802751]  
Right: 280.26791945099154  
Intercept 279.79997822099864  
Prediction\_local [279.75347944]  
Right: 279.8345971378414  
Intercept 279.8633843304392  
Prediction\_local [279.5705522]  
Right: 279.61772587566475  
Intercept 279.8180163687431  
Prediction\_local [279.3977172]  
Right: 279.3174486919262  
Intercept 279.96154384684604  
Prediction\_local [279.21361305]  
Right: 279.33440396032603  
Intercept 279.95488669321054  
Prediction\_local [279.29091168]  
Right: 279.1484398829653  
Intercept 279.8168922006962  
Prediction\_local [279.29319349]  
Right: 279.20559334786964  
Intercept 279.8717525691874  
Prediction\_local [279.24276116]  
Right: 279.550474883391  
Intercept 279.6853485513721  
Prediction\_local [279.68969751]  
Right: 279.7373287655614  
Intercept 279.7536342470741  
Prediction\_local [279.84784587]  
Right: 279.87692171821396  
Intercept 279.59073753809656  
Prediction\_local [280.28509793]  
Right: 280.37526702607585  
Intercept 279.5818290210784  
Prediction\_local [280.2159922]  
Right: 280.4688002914879  
Intercept 279.62138422383765  
Prediction\_local [280.16636587]  
Right: 280.3324786678442  
Intercept 279.7354185606654  
Prediction\_local [279.7911765]  
Right: 279.890903646998  
Intercept 279.74282112349135  
Prediction\_local [279.70619864]  
Right: 279.62016996410836  
Intercept 279.9422836671711  
Prediction\_local [279.52289005]  
Right: 279.59000595701565  
Intercept 279.75503199622875  
Prediction\_local [279.49773978]  
Right: 279.55827138723834  
Intercept 279.88260805305305  
Prediction\_local [279.26255528]  
Right: 278.794435909674  
Intercept 279.80019853648287  
Prediction\_local [279.40293219]  
Right: 278.9180227765941  
Intercept 279.86648146844027  
Prediction\_local [279.41289351]  
Right: 279.5086228520678  
Intercept 279.6678071742242  
Prediction\_local [279.77218333]  
Right: 279.8822871514032  
Intercept 279.63639429465326  
Prediction\_local [279.94503583]  
Right: 280.06749085780535  
Intercept 279.6951474967177  
Prediction\_local [280.06586677]  
Right: 280.2102420015299  
Intercept 279.4939047577384  
Prediction\_local [280.17481068]  
Right: 280.28929309662414  
Intercept 279.6987959892688  
Prediction\_local [280.03556354]  
Right: 280.31877229194475  
Intercept 279.72579031745937  
Prediction\_local [279.88162694]  
Right: 279.9593512599853  
Intercept 279.77917191356295  
Prediction\_local [279.74531697]  
Right: 279.73078269232855  
Intercept 279.7994230176439  
Prediction\_local [279.56917886]  
Right: 279.60317635815215  
Intercept 279.7566526981312  
Prediction\_local [279.56897954]  
Right: 279.5241421135817  
Intercept 279.96471450884445  
Prediction\_local [279.20593127]  
Right: 279.048009186407  
Intercept 279.83672789408456  
Prediction\_local [279.47296497]  
Right: 279.5795268606576  
Intercept 279.9140538642068  
Prediction\_local [279.33807325]  
Right: 279.6453732452038  
Intercept 279.75574640029026  
Prediction\_local [279.63272558]  
Right: 279.71312898562184  
Intercept 279.67452585447285  
Prediction\_local [279.96398369]  
Right: 279.91873053655985  
Intercept 279.5648898532701  
Prediction\_local [280.1905548]  
Right: 280.35254120976316

exp

<lime.explanation.Explanation at 0x1d582c83f40>

### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()

# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_R.columns);  
 plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_AT <= 26.86', 0.42825273577660267),  
 ('R\_P <= 0.05', 0.34599524994924263),  
 ('0.79 < R\_E <= 7.12', -0.14858303680657686)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.42825273577660267),  
 (2, -0.34599524994924263),  
 (0, 0.14858303680657686)],  
 1: [(1, 0.42825273577660267),  
 (2, 0.34599524994924263),  
 (0, -0.14858303680657686)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.1905548]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.35254120976316

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_RG\_Whole.html')

###########################################################################################################

###### Training and fitting the linear model on the training dataset  
lm3 = model.fit(X\_R\_train, y\_G\_train)

# The training data metrics

# The R-sq on the trainong dataset  
  
print('The training R2 is:', lm3.score(X\_R\_train, y\_G\_train))

The training R2 is: 0.5439704851909416

#### Predicting on the training dataset  
lm3\_predtr = lm3.predict(X\_R\_train)

# The training MAE  
mae\_tr = mean\_absolute\_error(y\_G\_train, lm3\_predtr)

print('The training MAE is:', mae\_tr)

The training MAE is: 0.2962090361587563

# The training MSE  
mse\_tr = mean\_squared\_error(y\_G\_train, lm3\_predtr)

print('The training MSE is:', mse\_tr)

The training MSE is: 0.13904164056098103

####### Predicting on the testing dataset  
  
lm3\_predts = lm3.predict(X\_R\_test)

# The testing data metrics

# The R-sq on the testing dataset  
print('The testing R2 is:', lm3.score(X\_R\_test, y\_G\_test))

The testing R2 is: 0.28918591536866645

# The testing MAE  
mae\_ts = mean\_absolute\_error(y\_G\_test, lm3\_predts)

print('The testing MAE is:', mae\_ts)

The testing MAE is: 0.3426910664039261

##### The Testing MSE  
mse\_ts = mean\_squared\_error(y\_G\_test, lm3\_predts)

#### Printing the MSE  
print(' The Testing MSE is:', mse\_ts)

The Testing MSE is: 0.19492803826048033

## The k-fold Cross-validation  
import numpy as np

# On the training dataset  
score\_train = cross\_val\_score(lm3, X\_R\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_train

array([-0.23143887, -0.12595127, -0.11500475, -0.17156583, -0.14893529,  
 -0.31159303, -0.10077905, -0.09696684, -0.11538802, -0.12602348])

print('The training CV MSE is:', absolute(np.mean(score\_train)))

The training CV MSE is: 0.15436464094712737

## On the testing dataset  
score\_test = cross\_val\_score(lm3, X\_R\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

print('The testing CV MSE is:', absolute(np.mean(score\_test)))

The testing CV MSE is: 0.18797299573913553

# Explain individual predictionsusing LimeTabularExplainer

# 1. Creating Explainer object

###### Initializing the explainer  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_R\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the full training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, lm3.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 279.60645296399804  
Prediction\_local [279.79902885]  
Right: 279.7822605764409  
Intercept 279.8262215246181  
Prediction\_local [279.52038523]  
Right: 279.7028496445306  
Intercept 279.52108706688045  
Prediction\_local [280.29786627]  
Right: 280.12613510078063  
Intercept 279.94173848088565  
Prediction\_local [279.01876544]  
Right: 279.1086784254148  
Intercept 279.8795958981761  
Prediction\_local [279.35098305]  
Right: 278.82608678464345  
Intercept 279.584047588866  
Prediction\_local [279.95693082]  
Right: 280.00313888694524  
Intercept 279.90682145737594  
Prediction\_local [279.07815504]  
Right: 278.58154941719044  
Intercept 279.82142107507804  
Prediction\_local [279.41376995]  
Right: 279.6831083517763  
Intercept 279.80512788538294  
Prediction\_local [279.37001509]  
Right: 279.5420093304609  
Intercept 279.88723975534526  
Prediction\_local [279.07310654]  
Right: 279.04911996601885  
Intercept 279.84413542581797  
Prediction\_local [279.04690593]  
Right: 279.54656311178655  
Intercept 279.7063356477951  
Prediction\_local [279.5359796]  
Right: 279.5475609068187  
Intercept 279.5492270839907  
Prediction\_local [279.96048106]  
Right: 279.841011676476  
Intercept 279.750223502151  
Prediction\_local [279.4057304]  
Right: 279.5410040915061  
Intercept 279.49418268369976  
Prediction\_local [280.16431244]  
Right: 280.3079469804505  
Intercept 279.569551210054  
Prediction\_local [279.72155419]  
Right: 279.32140412370507  
Intercept 279.56804292325035  
Prediction\_local [280.28474826]  
Right: 280.0498730118851  
Intercept 279.55986677358527  
Prediction\_local [279.84187377]  
Right: 279.62280861401143  
Intercept 279.7707123342879  
Prediction\_local [279.43925095]  
Right: 279.57640848213697  
Intercept 279.7903197291359  
Prediction\_local [279.32223816]  
Right: 279.3514006269549  
Intercept 279.73703265761776  
Prediction\_local [279.47836752]  
Right: 279.2580385653762  
Intercept 279.5826009242007  
Prediction\_local [280.26981239]  
Right: 280.3491698657389  
Intercept 279.6406951212174  
Prediction\_local [279.79981004]  
Right: 279.74112559546745  
Intercept 279.9094270454798  
Prediction\_local [279.24757152]  
Right: 279.602330192785  
Intercept 279.6497069735774  
Prediction\_local [279.41821845]  
Right: 279.74195951694566  
Intercept 279.51235460493757  
Prediction\_local [280.34036591]  
Right: 280.2544809141094  
Intercept 279.6807552789067  
Prediction\_local [279.63266118]  
Right: 279.66616074165455  
Intercept 279.5019152993816  
Prediction\_local [280.15599608]  
Right: 280.26277698541514  
Intercept 279.8754137383762  
Prediction\_local [279.03801803]  
Right: 279.08121758673326  
Intercept 279.78001070577926  
Prediction\_local [279.73376119]  
Right: 279.90067682349473  
Intercept 279.50263911799294  
Prediction\_local [279.64910365]  
Right: 279.47049356240376  
Intercept 279.58853729896555  
Prediction\_local [280.13811652]  
Right: 280.04087573328457  
Intercept 279.5444044637328  
Prediction\_local [280.23321536]  
Right: 280.49404125042577  
Intercept 279.5604576953857  
Prediction\_local [280.15883602]  
Right: 280.03885115122466  
Intercept 279.41703869979324  
Prediction\_local [280.32094239]  
Right: 280.2602723924311  
Intercept 279.82036702308227  
Prediction\_local [279.18194334]  
Right: 279.38999058835594  
Intercept 279.4367137919152  
Prediction\_local [280.32245858]  
Right: 280.2316392576126  
Intercept 279.60692912574297  
Prediction\_local [279.88229121]  
Right: 280.11837788154315  
Intercept 279.8195236164355  
Prediction\_local [279.50236172]  
Right: 279.5222327538229  
Intercept 279.7172736821469  
Prediction\_local [279.5760706]  
Right: 279.75530959210454  
Intercept 279.67811907411203  
Prediction\_local [279.7183917]  
Right: 279.8913498092822  
Intercept 279.4720769784707  
Prediction\_local [280.13639352]  
Right: 279.71167834072975  
Intercept 279.64567123080565  
Prediction\_local [279.89754149]  
Right: 279.7293621248524  
Intercept 279.5380743948629  
Prediction\_local [279.9823288]  
Right: 279.93213703943013  
Intercept 279.7653326154985  
Prediction\_local [279.39833296]  
Right: 279.6404968446552  
Intercept 279.9196948083979  
Prediction\_local [279.13792726]  
Right: 278.91719093426326  
Intercept 279.65716369042866  
Prediction\_local [279.94994681]  
Right: 279.88055449831586  
Intercept 279.6752053404636  
Prediction\_local [279.77652687]  
Right: 279.9341302267289  
Intercept 279.53150097924004  
Prediction\_local [280.00346941]  
Right: 280.0044391792378  
Intercept 279.80179754400905  
Prediction\_local [279.3872552]  
Right: 278.89485579989906  
Intercept 279.8226416036553  
Prediction\_local [279.53796131]  
Right: 279.65175771157186  
Intercept 279.5316896146098  
Prediction\_local [279.99789806]  
Right: 280.1405699208363  
Intercept 279.4554792366227  
Prediction\_local [280.27970258]  
Right: 280.44124663626536  
Intercept 279.6397603138427  
Prediction\_local [279.51477387]  
Right: 279.3940538860064  
Intercept 279.71490798097733  
Prediction\_local [279.41546727]  
Right: 279.4897135953852  
Intercept 279.5205181302598  
Prediction\_local [280.00089979]  
Right: 279.98553036016574  
Intercept 279.61709264167445  
Prediction\_local [279.64797688]  
Right: 279.75867189672016  
Intercept 279.64542167062626  
Prediction\_local [279.64376794]  
Right: 279.6821034710176  
Intercept 279.81246506370803  
Prediction\_local [279.42528304]  
Right: 279.49419107800594  
Intercept 279.57955835764886  
Prediction\_local [279.73926714]  
Right: 279.5833055093756  
Intercept 279.7542638612105  
Prediction\_local [279.53965478]  
Right: 279.1999364141025  
Intercept 279.8381906768714  
Prediction\_local [279.58717923]  
Right: 279.5609973686512  
Intercept 279.8204734344007  
Prediction\_local [279.06663322]  
Right: 279.29550150577495  
Intercept 279.4759466237804  
Prediction\_local [280.1608049]  
Right: 280.3880454429481  
Intercept 279.82037930689023  
Prediction\_local [279.88258381]  
Right: 279.903819634042  
Intercept 279.86754044028015  
Prediction\_local [279.41346633]  
Right: 279.6696642208298  
Intercept 279.5010133992959  
Prediction\_local [280.12664605]  
Right: 280.3232416855387  
Intercept 279.9991769949066  
Prediction\_local [278.72166518]  
Right: 279.11751969205926  
Intercept 279.7602821552034  
Prediction\_local [279.46184521]  
Right: 279.15891916414216  
Intercept 279.9754034146889  
Prediction\_local [278.82062825]  
Right: 279.2335629555241  
Intercept 279.7761465628576  
Prediction\_local [279.53457166]  
Right: 279.6291850500159  
Intercept 279.5107448527446  
Prediction\_local [279.91427681]  
Right: 279.9472642123003  
Intercept 279.89917406062347  
Prediction\_local [279.55687102]  
Right: 279.58744964636276  
Intercept 279.88216514576624  
Prediction\_local [279.38330085]  
Right: 279.3804016229782  
Intercept 279.78204257202907  
Prediction\_local [279.77629414]  
Right: 279.69087666237635  
Intercept 279.54268954564935  
Prediction\_local [280.31926804]  
Right: 280.1426804236972  
Intercept 279.54688727288965  
Prediction\_local [280.08035894]  
Right: 280.1180830364891  
Intercept 279.80388212720175  
Prediction\_local [279.69238339]  
Right: 279.73347464354254  
Intercept 279.7986023137217  
Prediction\_local [279.64844334]  
Right: 279.76177978729766  
Intercept 279.3394154535365  
Prediction\_local [280.16920262]  
Right: 280.32984522856685  
Intercept 279.59684578350686  
Prediction\_local [280.05900444]  
Right: 280.23947367915486

exp

<lime.explanation.Explanation at 0x25d8d8b00a0>

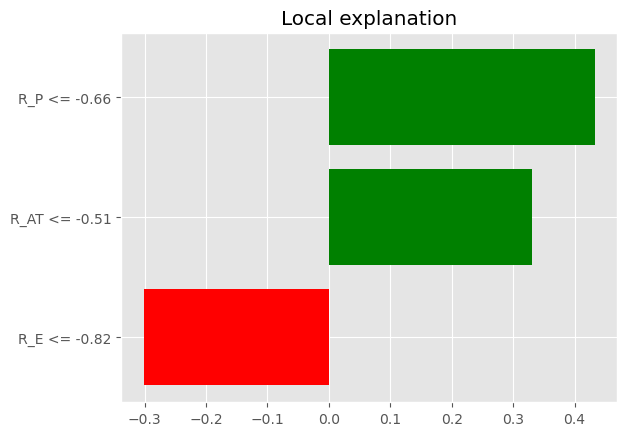
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

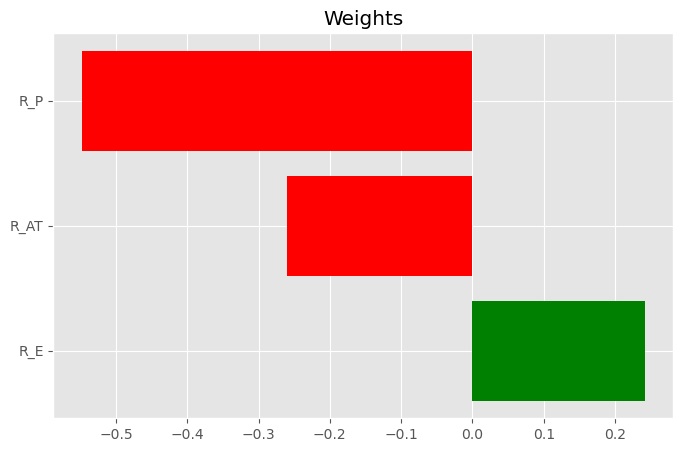
##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_R\_train.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P <= -0.66', 0.4331653258956005),  
 ('R\_AT <= -0.51', 0.330052648847184),  
 ('R\_E <= -0.82', -0.3010593139967057)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.4331653258956005),  
 (1, -0.330052648847184),  
 (0, 0.3010593139967057)],  
 1: [(2, 0.4331653258956005),  
 (1, 0.330052648847184),  
 (0, -0.3010593139967057)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.05900444]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.23947367915486

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_RG\_train.html')

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, lm3.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.59937549572487  
Prediction\_local [279.74469202]  
Right: 279.88945494869677  
Intercept 279.7260649209071  
Prediction\_local [280.09258978]  
Right: 280.03321535329513  
Intercept 279.88852196946067  
Prediction\_local [279.73349217]  
Right: 279.56564014929336  
Intercept 279.67901221688874  
Prediction\_local [280.07775785]  
Right: 280.0368826875921  
Intercept 280.1109393311841  
Prediction\_local [279.58837774]  
Right: 279.4720621880949  
Intercept 279.65771878557695  
Prediction\_local [280.11380016]  
Right: 280.0033404601686  
Intercept 279.6622335283671  
Prediction\_local [280.14468282]  
Right: 279.722141817862  
Intercept 279.507757188635  
Prediction\_local [280.15023835]  
Right: 280.28334386309825  
Intercept 279.7549008281612  
Prediction\_local [279.12818367]  
Right: 278.8010633613697  
Intercept 279.2424756245152  
Prediction\_local [280.15440742]  
Right: 280.33499308836815  
Intercept 279.6937999401629  
Prediction\_local [279.42534921]  
Right: 279.56865121340667  
Intercept 279.53781702140174  
Prediction\_local [279.76817263]  
Right: 279.98458530068996  
Intercept 279.5178211895715  
Prediction\_local [280.0674804]  
Right: 280.2843461425685  
Intercept 279.31179293574576  
Prediction\_local [279.89911346]  
Right: 279.88051286986024  
Intercept 279.72435535610816  
Prediction\_local [279.6839904]  
Right: 279.62311746208  
Intercept 279.61227729458557  
Prediction\_local [279.74173525]  
Right: 279.56483154391134  
Intercept 279.7577105692227  
Prediction\_local [279.9667633]  
Right: 279.81912117941715  
Intercept 279.65113545948526  
Prediction\_local [280.07255674]  
Right: 280.0967123689614  
Intercept 279.9393958370685  
Prediction\_local [279.21650338]  
Right: 279.05533192203137  
Intercept 279.9333072876296  
Prediction\_local [279.0370042]  
Right: 278.6475992942018  
Intercept 279.590137470903  
Prediction\_local [279.50675781]  
Right: 279.9149433948812  
Intercept 279.8476008311454  
Prediction\_local [279.23195665]  
Right: 279.11282005249603  
Intercept 279.6732570117835  
Prediction\_local [280.14802615]  
Right: 280.38155668422763  
Intercept 279.7521919924718  
Prediction\_local [280.29519495]  
Right: 280.29888725573886  
Intercept 279.5117673530919  
Prediction\_local [280.06515779]  
Right: 280.174686200452  
Intercept 279.8392362638329  
Prediction\_local [279.58696331]  
Right: 279.55473180594976  
Intercept 279.7904999680372  
Prediction\_local [279.16557744]  
Right: 278.59494752817767

exp

<lime.explanation.Explanation at 0x25d8d913ca0>

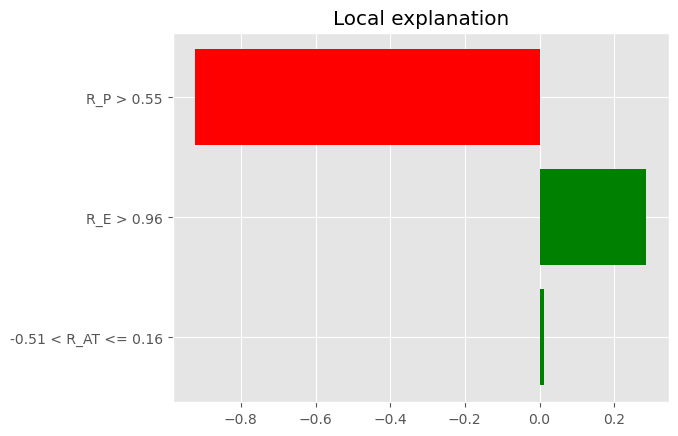
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

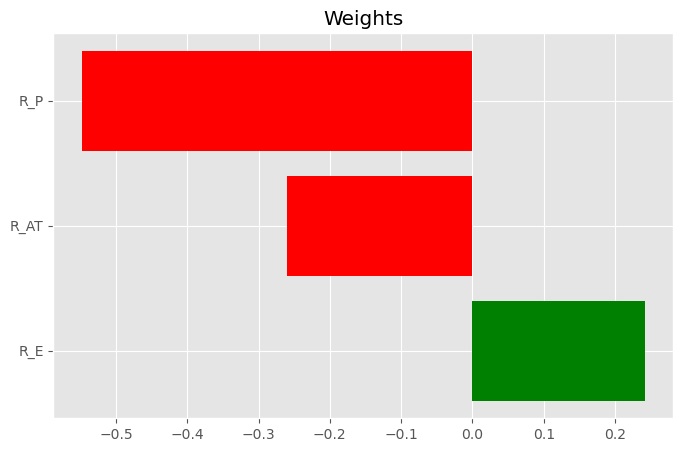
##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_R\_test.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P > 0.55', -0.921746382561321),  
 ('R\_E > 0.96', 0.2860432349679614),  
 ('-0.51 < R\_AT <= 0.16', 0.01078061549535839)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 0.921746382561321),  
 (0, -0.2860432349679614),  
 (1, -0.01078061549535839)],  
 1: [(2, -0.921746382561321),  
 (0, 0.2860432349679614),  
 (1, 0.01078061549535839)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.16557744]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 278.59494752817767

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_RG\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

############ MODEL 4: GT Features (X\_G) and GT Target (LL\_G)###############

###### Training and fitting the linear model on the whole dataset

# Instantiation using LinearRegression  
model = LinearRegression(fit\_intercept = True, copy\_X = True, positive = False, n\_jobs = None)

###### Training and fiting the linear model on the whole dataset  
lm4 = model.fit(X\_G, y\_G)

# The model interept  
print('The intercept is:', lm4.intercept\_)

The intercept is: 282.35183666863077

# The model coefficients  
print('The model coefficients are:', lm4.coef\_)

The model coefficients are: [ 0.00139905 -0.08698268 -0.00360961]

################ Finding metrics on the whole dataset

# The coefficient of determination  
  
print('The coefficient of determination is:', lm4.score(X\_G, y\_G))

The coefficient of determination is: 0.5292989785047095

# Making the prediction  
lm4\_pred = lm4.predict(X\_G)

# lm1 MAE  
print('The lm4 MAE is:%.2f'% mean\_absolute\_error(y\_G, lm4\_pred))

The lm4 MAE is:0.30

# lm1 MSE  
print('The lm4 MSE is:%.2f'% mean\_squared\_error(y\_G, lm4\_pred))

The lm4 MSE is:0.14

## The k-fold Cross-validation  
import numpy as np

# On the full dataset  
score\_lm4 = cross\_val\_score(lm4, X\_G, y\_G, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_lm4

array([-0.05157151, -0.07499733, -0.24434023, -0.36193465, -0.13214074,  
 -0.05297778, -0.10524647, -0.27509414, -0.23071209, -0.12305856])

# The absolute mean score on the full dataset

print('The CV MSE is:', absolute(np.mean(score\_lm4)))

The CV MSE is: 0.16520734915166288

# LIME

###### Initializing the explainer  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_R\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the full XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, lm4.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 279.46985284330634  
Prediction\_local [280.33767867]  
Right: 280.4081451227053  
Intercept 279.5056708638979  
Prediction\_local [280.2998676]  
Right: 280.2477867792772  
Intercept 279.66810134471285  
Prediction\_local [279.43611863]  
Right: 279.597527303523  
Intercept 279.7405596773241  
Prediction\_local [279.55312969]  
Right: 279.4840300992799  
Intercept 279.75073207760136  
Prediction\_local [279.54134163]  
Right: 279.52963209125187  
Intercept 279.9299559785083  
Prediction\_local [279.4853224]  
Right: 279.5478565786033  
Intercept 279.922988634294  
Prediction\_local [279.23351657]  
Right: 279.2151399576945  
Intercept 279.99090274338124  
Prediction\_local [279.21381651]  
Right: 279.35556025360813  
Intercept 279.6530539168793  
Prediction\_local [279.89361896]  
Right: 279.7110955030677  
Intercept 279.6222645427497  
Prediction\_local [279.95181061]  
Right: 279.7362919528814  
Intercept 279.51279124348184  
Prediction\_local [280.29228012]  
Right: 280.17877232503895  
Intercept 279.43337121796395  
Prediction\_local [280.34354743]  
Right: 280.4547264513985  
Intercept 279.40141639987735  
Prediction\_local [280.36686588]  
Right: 280.47408166153406  
Intercept 279.4796704620829  
Prediction\_local [280.31498876]  
Right: 280.4008459068176  
Intercept 279.6186499242701  
Prediction\_local [279.68597237]  
Right: 279.6924966575551  
Intercept 279.69130534696285  
Prediction\_local [279.61105083]  
Right: 279.49706543024894  
Intercept 279.7730609868261  
Prediction\_local [279.55112839]  
Right: 279.5390178142771  
Intercept 279.69680177435123  
Prediction\_local [279.55808974]  
Right: 279.5299119017436  
Intercept 279.9451372506064  
Prediction\_local [279.20686828]  
Right: 279.2325697197932  
Intercept 279.86584158034776  
Prediction\_local [279.34796387]  
Right: 279.0816159198033  
Intercept 279.80817182488454  
Prediction\_local [279.69121021]  
Right: 279.6208042587513  
Intercept 279.6852595908485  
Prediction\_local [279.91821263]  
Right: 279.75270915285336  
Intercept 279.603274492511  
Prediction\_local [279.8574985]  
Right: 280.13626032002725  
Intercept 279.46405313151257  
Prediction\_local [280.33918579]  
Right: 280.49021905101426  
Intercept 279.4208808924409  
Prediction\_local [280.32402342]  
Right: 280.40070600157173  
Intercept 279.5204257493517  
Prediction\_local [280.32502073]  
Right: 280.1969962216566  
Intercept 279.7555504011578  
Prediction\_local [279.56000213]  
Right: 279.61014291875443  
Intercept 279.6412986795812  
Prediction\_local [279.55965576]  
Right: 279.53831828804766  
Intercept 279.6342209354109  
Prediction\_local [279.50065157]  
Right: 279.51588516250257  
Intercept 279.92130547614715  
Prediction\_local [279.28897495]  
Right: 279.4997085963909  
Intercept 279.7443041400813  
Prediction\_local [279.59896771]  
Right: 279.5562056820624  
Intercept 279.865542738427  
Prediction\_local [279.46296422]  
Right: 279.2415971099647  
Intercept 279.77890230384827  
Prediction\_local [279.58570372]  
Right: 279.5934230929391  
Intercept 279.6264973614443  
Prediction\_local [279.85597553]  
Right: 279.75101821835193  
Intercept 279.4918071665079  
Prediction\_local [279.82817834]  
Right: 279.960883829086  
Intercept 279.4025149039916  
Prediction\_local [280.25884606]  
Right: 280.48375926660185  
Intercept 279.39653553626295  
Prediction\_local [280.34428824]  
Right: 280.53664840291077  
Intercept 279.47739349470254  
Prediction\_local [280.30572624]  
Right: 280.5676519631073  
Intercept 279.6869090441049  
Prediction\_local [279.6794763]  
Right: 279.67228994439347  
Intercept 279.6448394656417  
Prediction\_local [279.61590017]  
Right: 279.4542856862963  
Intercept 279.8971254920446  
Prediction\_local [279.47787191]  
Right: 279.4364391954811  
Intercept 279.85164598513904  
Prediction\_local [279.28347491]  
Right: 279.4846869796017  
Intercept 279.93305842550086  
Prediction\_local [279.31868304]  
Right: 279.41977612462256  
Intercept 279.85413735443564  
Prediction\_local [279.56661885]  
Right: 279.1524298795993  
Intercept 279.8854080450934  
Prediction\_local [279.70485846]  
Right: 279.7544514818195  
Intercept 279.68705500158177  
Prediction\_local [279.71067685]  
Right: 279.7212858769191  
Intercept 279.6103304946013  
Prediction\_local [279.90800447]  
Right: 279.8793936648623  
Intercept 279.4880324573813  
Prediction\_local [280.23907169]  
Right: 280.2078051402423  
Intercept 279.5058299826339  
Prediction\_local [280.29487967]  
Right: 280.2623731395018  
Intercept 279.46831560606546  
Prediction\_local [280.29641072]  
Right: 280.2211202817031  
Intercept 279.56525031055344  
Prediction\_local [279.93676576]  
Right: 279.8289986798777  
Intercept 279.67549563376656  
Prediction\_local [279.54714556]  
Right: 279.4976250512325  
Intercept 279.7410994994907  
Prediction\_local [279.50178194]  
Right: 279.538622241641  
Intercept 279.8624551675225  
Prediction\_local [279.57308986]  
Right: 279.56111973709125  
Intercept 279.89797184936356  
Prediction\_local [279.43364724]  
Right: 279.3653544640897  
Intercept 279.90061622611233  
Prediction\_local [279.35822536]  
Right: 279.3345717071974  
Intercept 279.75335449589835  
Prediction\_local [279.69608144]  
Right: 279.58980069855477  
Intercept 279.8349796698422  
Prediction\_local [279.62164156]  
Right: 279.7092178111205  
Intercept 279.4593527288795  
Prediction\_local [279.87578225]  
Right: 280.04675934213554  
Intercept 279.51383240068435  
Prediction\_local [280.2909055]  
Right: 280.31120502368  
Intercept 279.40548949610627  
Prediction\_local [280.28510873]  
Right: 280.2960348993702  
Intercept 279.6457118726482  
Prediction\_local [279.96391005]  
Right: 280.1094418456565  
Intercept 279.721334401432  
Prediction\_local [279.71561693]  
Right: 279.65952235236534  
Intercept 279.6751747077431  
Prediction\_local [279.47092409]  
Right: 279.4298697442072  
Intercept 279.57694991810916  
Prediction\_local [279.53246171]  
Right: 279.51182791037184  
Intercept 279.9319105169617  
Prediction\_local [279.18554583]  
Right: 279.2604789198637  
Intercept 279.8476298055066  
Prediction\_local [279.32723259]  
Right: 279.3599308798342  
Intercept 279.7866600167503  
Prediction\_local [279.43005874]  
Right: 279.00870539600083  
Intercept 279.87871058845366  
Prediction\_local [279.57779503]  
Right: 279.6265427723457  
Intercept 279.64930951405546  
Prediction\_local [279.98778868]  
Right: 280.119413249025  
Intercept 279.60412820184746  
Prediction\_local [279.9710975]  
Right: 280.0367898550252  
Intercept 279.445164543004  
Prediction\_local [280.36288754]  
Right: 280.20303629033134  
Intercept 279.4083825353236  
Prediction\_local [280.32648894]  
Right: 280.2820081601291  
Intercept 279.4070388403492  
Prediction\_local [280.32826065]  
Right: 280.1867469240545  
Intercept 279.6175629482621  
Prediction\_local [279.70355719]  
Right: 279.6624724340798  
Intercept 279.6626606576863  
Prediction\_local [279.52978842]  
Right: 279.49692552500306  
Intercept 279.7406236031062  
Prediction\_local [279.64132918]  
Right: 279.473233252245  
Intercept 279.9126677552355  
Prediction\_local [279.17201411]  
Right: 279.4212826979657  
Intercept 279.9173638389221  
Prediction\_local [279.2842429]  
Right: 279.4969513924472  
Intercept 279.9792789037752  
Prediction\_local [279.26013932]  
Right: 279.0371091123591  
Intercept 279.7492954056179  
Prediction\_local [279.65285696]  
Right: 279.6211745569026  
Intercept 279.8101620130848  
Prediction\_local [279.5721345]  
Right: 279.5237987340447  
Intercept 279.6042100553797  
Prediction\_local [279.85458909]  
Right: 280.0307618579012  
Intercept 279.4089344814501  
Prediction\_local [280.30136076]  
Right: 280.1992467771416  
Intercept 279.4669008382191  
Prediction\_local [280.32371373]  
Right: 280.39522555388055  
Intercept 279.5121994514336  
Prediction\_local [280.22540468]  
Right: 280.40364401173537  
Intercept 279.6736316438166  
Prediction\_local [279.69519199]  
Right: 279.70119492590175  
Intercept 279.72206393344254  
Prediction\_local [279.54593977]  
Right: 279.44964467008043  
Intercept 279.7243622489421  
Prediction\_local [279.52478074]  
Right: 279.53707121238546  
Intercept 279.8515632311833  
Prediction\_local [279.59734264]  
Right: 279.55891332213724  
Intercept 279.88578643795057  
Prediction\_local [279.35012946]  
Right: 279.46800111398255  
Intercept 279.92226240783225  
Prediction\_local [279.26794283]  
Right: 279.2220658884097  
Intercept 279.7920250068232  
Prediction\_local [279.48637678]  
Right: 279.48432202135155  
Intercept 279.6786597972496  
Prediction\_local [279.92697227]  
Right: 279.7960726608911  
Intercept 279.62701180260666  
Prediction\_local [279.89573964]  
Right: 280.0255732922526  
Intercept 279.4302765833203  
Prediction\_local [280.34522743]  
Right: 280.38500039938003  
Intercept 279.4672769529296  
Prediction\_local [280.36416107]  
Right: 280.3942462171593  
Intercept 279.44398729318107  
Prediction\_local [280.33726574]  
Right: 280.32858948882233  
Intercept 279.6726348383467  
Prediction\_local [279.60348413]  
Right: 279.6394676160004  
Intercept 279.7471554807256  
Prediction\_local [279.53378868]  
Right: 279.50071503819277  
Intercept 279.7011608980247  
Prediction\_local [279.50613252]  
Right: 279.56161498816965  
Intercept 279.720929251701  
Prediction\_local [279.60452511]  
Right: 279.6327734037149  
Intercept 279.9271806647853  
Prediction\_local [279.37428477]  
Right: 279.1781225928599  
Intercept 279.97523103511105  
Prediction\_local [279.22019871]  
Right: 279.4360886216325  
Intercept 279.7892774896317  
Prediction\_local [279.49629878]  
Right: 279.2337771137524  
Intercept 279.62820219425527  
Prediction\_local [279.89912977]  
Right: 279.75047066891915  
Intercept 279.619309029931  
Prediction\_local [279.90935192]  
Right: 279.97661357437937  
Intercept 279.44465398103995  
Prediction\_local [280.3522257]  
Right: 280.3507669469772

exp

<lime.explanation.Explanation at 0x1d582c2ffd0>

### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()

# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_G.columns);  
 plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= 25.38', 0.6046784058225562),  
 ('G\_P <= 0.00', 0.2716010788096635),  
 ('9.70 < G\_E <= 13.62', 0.03129222978557029)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.6046784058225562),  
 (2, -0.2716010788096635),  
 (0, -0.03129222978557029)],  
 1: [(1, 0.6046784058225562),  
 (2, 0.2716010788096635),  
 (0, 0.03129222978557029)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.3522257]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.3507669469772

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_GG\_Whole.html')

###########################################################################################################

###### Training and fitting the linear model on the training dataset  
lm4 = model.fit(X\_G\_train, y\_G\_train)

# The training data metrics

# The R-sq on the trainong dataset  
  
print('The training R2 is:', lm4.score(X\_G\_train, y\_G\_train))

The training R2 is: 0.539087704060961

#### Predicting on the training dataset  
lm4\_predtr = lm4.predict(X\_G\_train)

# The training MAE  
mae\_tr = mean\_absolute\_error(y\_G\_train, lm4\_predtr)

print('The training MAE is:', mae\_tr)

The training MAE is: 0.2964986183338297

# The training MSE  
mse\_tr = mean\_squared\_error(y\_G\_train, lm4\_predtr)

print('The training MSE is:', mse\_tr)

The training MSE is: 0.14053038169892465

####### Predicting on the testing dataset  
  
lm4\_predts = lm4.predict(X\_G\_test)

# The testing data metrics

# The R-sq on the testing dataset  
print('The testing R2 is:', lm4.score(X\_G\_test, y\_G\_test))

The testing R2 is: 0.33551373578528343

# The testing MAE  
mae\_ts = mean\_absolute\_error(y\_G\_test, lm4\_predts)

print('The testing MAE is:', mae\_ts)

The testing MAE is: 0.31579849728203474

##### The Testing MSE  
mse\_ts = mean\_squared\_error(y\_G\_test, lm4\_predts)

#### Printing the MSE  
print(' The Testing MSE is:', mse\_ts)

The Testing MSE is: 0.18222346283640342

## The k-fold Cross-validation  
import numpy as np

# On the training dataset  
score\_train = cross\_val\_score(lm4, X\_G\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_train

array([-0.1023142 , -0.12911317, -0.14871866, -0.14980839, -0.12332414,  
 -0.27614414, -0.16905776, -0.05476693, -0.17030705, -0.22172011])

print('The training CV MSE is:', absolute(np.mean(score\_train)))

The training CV MSE is: 0.15452745416915384

## On the testing dataset  
score\_test = cross\_val\_score(lm4, X\_G\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

print('The testing CV MSE is:', absolute(np.mean(score\_test)))

The testing CV MSE is: 0.17079422022377347

# Explain individual predictionsusing LimeTabularExplainer

# 1. Creating Explainer object

###### Initializing the explainer  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

#### Explanation of a prediction  
# i = 10 ### Index of the instance to be explained

# exp = explainer.explain\_instance(X\_R\_test.values[i], model.predict, num\_features = 3)

# Explanation of the prediction for the full training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, lm4.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 279.56335147314417  
Prediction\_local [279.48447991]  
Right: 279.6306185872822  
Intercept 279.66275881719133  
Prediction\_local [279.41000357]  
Right: 279.4789909548133  
Intercept 279.2980496542195  
Prediction\_local [280.16285677]  
Right: 280.2741554874581  
Intercept 279.8001448368792  
Prediction\_local [279.30736349]  
Right: 279.34037271248275  
Intercept 279.6596432874222  
Prediction\_local [279.48715214]  
Right: 279.01893664628085  
Intercept 279.30352792175154  
Prediction\_local [280.1669448]  
Right: 280.15644758775295  
Intercept 279.65079654794323  
Prediction\_local [279.45634612]  
Right: 279.38979908801895  
Intercept 279.60132015106933  
Prediction\_local [279.40479566]  
Right: 279.5365798791319  
Intercept 279.5748419137619  
Prediction\_local [279.45278705]  
Right: 279.56576651295836  
Intercept 279.65670299969776  
Prediction\_local [279.34668689]  
Right: 279.47018197290345  
Intercept 279.6026311908444  
Prediction\_local [279.69487721]  
Right: 279.63667313183475  
Intercept 279.6228820466572  
Prediction\_local [279.35851173]  
Right: 279.4170918390247  
Intercept 279.68864412052426  
Prediction\_local [279.45661504]  
Right: 279.6700557623175  
Intercept 279.79572547388614  
Prediction\_local [279.23720512]  
Right: 279.3566749620054  
Intercept 279.32883166306686  
Prediction\_local [280.12593363]  
Right: 280.16709030062896  
Intercept 279.7776066585013  
Prediction\_local [279.24878521]  
Right: 279.37345983319545  
Intercept 279.34540341366994  
Prediction\_local [279.7517212]  
Right: 279.9266684769471  
Intercept 279.6866980647914  
Prediction\_local [279.41644902]  
Right: 279.6199259365496  
Intercept 279.76272759791544  
Prediction\_local [279.23184742]  
Right: 279.4768688803489  
Intercept 279.68044318505997  
Prediction\_local [279.42134199]  
Right: 279.494329120142  
Intercept 279.82841754980655  
Prediction\_local [279.09322656]  
Right: 278.84244510373514  
Intercept 279.2118489201613  
Prediction\_local [280.1030189]  
Right: 280.16060779708977  
Intercept 279.5707038888132  
Prediction\_local [279.49800198]  
Right: 279.6616738720433  
Intercept 279.607116216762  
Prediction\_local [279.46227829]  
Right: 279.46964400327147  
Intercept 279.6128946978239  
Prediction\_local [279.50354527]  
Right: 279.5809001352377  
Intercept 279.4265990590788  
Prediction\_local [280.15097102]  
Right: 280.33011425350776  
Intercept 279.7252948759547  
Prediction\_local [279.411973]  
Right: 279.5690903048898  
Intercept 279.3221224033832  
Prediction\_local [280.15267374]  
Right: 280.2101147954005  
Intercept 279.78292823016653  
Prediction\_local [279.11116474]  
Right: 279.25516905605895  
Intercept 279.34769476476447  
Prediction\_local [280.16212676]  
Right: 280.1478235079448  
Intercept 279.64718203084766  
Prediction\_local [279.48005631]  
Right: 279.4569901662716  
Intercept 279.36234523609505  
Prediction\_local [280.13616733]  
Right: 280.1105279091785  
Intercept 279.42129713613826  
Prediction\_local [280.13650456]  
Right: 280.4729985598356  
Intercept 279.57340138430425  
Prediction\_local [279.70166995]  
Right: 280.01377473515873  
Intercept 279.4648803907103  
Prediction\_local [280.10256758]  
Right: 280.16956257696444  
Intercept 279.6373046261436  
Prediction\_local [279.61303852]  
Right: 279.61444625048546  
Intercept 279.42433151278044  
Prediction\_local [280.11645144]  
Right: 280.12379288052307  
Intercept 279.57034714985195  
Prediction\_local [279.75479845]  
Right: 279.7463385897197  
Intercept 279.4345334719365  
Prediction\_local [279.48852134]  
Right: 279.4491121850534  
Intercept 279.47844785815954  
Prediction\_local [280.15785341]  
Right: 280.1208125618456  
Intercept 279.50560700341094  
Prediction\_local [279.66252607]  
Right: 279.972257455455  
Intercept 279.58090369866255  
Prediction\_local [279.76276178]  
Right: 279.7574493440297  
Intercept 279.4460643785023  
Prediction\_local [279.64750506]  
Right: 279.7252031461009  
Intercept 279.4788462159323  
Prediction\_local [279.77586802]  
Right: 280.03702101924625  
Intercept 279.473543113977  
Prediction\_local [279.51186591]  
Right: 279.5268714917233  
Intercept 279.8123450480399  
Prediction\_local [279.24325594]  
Right: 279.2108674117081  
Intercept 279.4191647050264  
Prediction\_local [279.72609144]  
Right: 279.7520367499247  
Intercept 279.427555385744  
Prediction\_local [279.66427989]  
Right: 279.78471849608655  
Intercept 279.3359136017145  
Prediction\_local [280.15819496]  
Right: 280.3470894877898  
Intercept 279.73384700049996  
Prediction\_local [279.16250051]  
Right: 278.9598422348746  
Intercept 279.54422639862224  
Prediction\_local [279.37640937]  
Right: 279.47193556167355  
Intercept 279.3100625043799  
Prediction\_local [280.10403788]  
Right: 280.4124566769818  
Intercept 279.2847496458387  
Prediction\_local [280.10556288]  
Right: 280.32610402630405  
Intercept 279.7040263852984  
Prediction\_local [279.41209828]  
Right: 279.5434852901385  
Intercept 279.71010828194187  
Prediction\_local [279.50303784]  
Right: 279.3341841807587  
Intercept 279.50102126015764  
Prediction\_local [279.54533552]  
Right: 279.61481986800146  
Intercept 279.6340289868606  
Prediction\_local [279.35919571]  
Right: 279.52301124665274  
Intercept 279.76510316640406  
Prediction\_local [279.50454068]  
Right: 279.62069874613985  
Intercept 279.7437652394925  
Prediction\_local [279.09499747]  
Right: 279.0622046503324  
Intercept 279.67143530626333  
Prediction\_local [279.37237875]  
Right: 279.52014679865016  
Intercept 279.6876887977433  
Prediction\_local [279.43746834]  
Right: 279.4815247027493  
Intercept 279.61735394822637  
Prediction\_local [279.53842327]  
Right: 279.57268896834466  
Intercept 279.76221548173186  
Prediction\_local [279.10223519]  
Right: 279.1017623914465  
Intercept 279.42179862215306  
Prediction\_local [280.15077254]  
Right: 280.42557166619326  
Intercept 279.43626683850124  
Prediction\_local [279.75609119]  
Right: 280.04434564085193  
Intercept 279.62083119437625  
Prediction\_local [279.47889525]  
Right: 279.0097952069039  
Intercept 279.31252998055885  
Prediction\_local [280.1482788]  
Right: 280.1590391104213  
Intercept 279.7803300772861  
Prediction\_local [279.1696278]  
Right: 279.14271655789315  
Intercept 279.76046013378044  
Prediction\_local [279.35551896]  
Right: 279.235747907048  
Intercept 279.71720254973513  
Prediction\_local [279.2306074]  
Right: 279.4540163353036  
Intercept 279.49523332584005  
Prediction\_local [279.39556786]  
Right: 279.4813132490156  
Intercept 279.6412015593945  
Prediction\_local [279.76481165]  
Right: 279.886358448045  
Intercept 279.5606830040836  
Prediction\_local [279.37137844]  
Right: 279.53543409993085  
Intercept 279.7149874701745  
Prediction\_local [279.5140173]  
Right: 279.5395119879331  
Intercept 279.5569680081397  
Prediction\_local [279.54725389]  
Right: 279.5885591537782  
Intercept 279.32218324970006  
Prediction\_local [280.12785876]  
Right: 280.4431197900758  
Intercept 279.3655251406654  
Prediction\_local [280.10290275]  
Right: 280.42120000312275  
Intercept 279.634241585608  
Prediction\_local [279.68654374]  
Right: 279.7762788310801  
Intercept 279.4643894965666  
Prediction\_local [279.76006915]  
Right: 279.7417862087158  
Intercept 279.32844014443333  
Prediction\_local [280.14802073]  
Right: 280.3101553248902  
Intercept 279.3530938776623  
Prediction\_local [280.12223632]  
Right: 280.5000436173931

exp

<lime.explanation.Explanation at 0x25d8f129670>

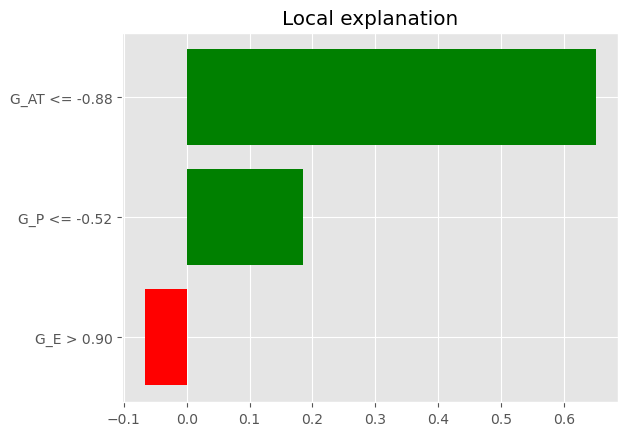
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_G\_train.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= -0.88', 0.6507828144844436),  
 ('G\_P <= -0.52', 0.18466228232117057),  
 ('G\_E > 0.90', -0.06630265871044516)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.6507828144844436),  
 (2, -0.18466228232117057),  
 (0, 0.06630265871044516)],  
 1: [(1, 0.6507828144844436),  
 (2, 0.18466228232117057),  
 (0, -0.06630265871044516)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.12223632]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.5000436173931

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_GG\_train.html')

# Explanation of the prediction for the full testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, lm4.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.4405595638343  
Prediction\_local [280.06707227]  
Right: 280.203843745595  
Intercept 279.32755172880894  
Prediction\_local [280.04923098]  
Right: 280.3612002740692  
Intercept 279.60319513504083  
Prediction\_local [279.49072663]  
Right: 279.4241164963641  
Intercept 279.57304963412656  
Prediction\_local [279.73029456]  
Right: 279.72685026844704  
Intercept 279.6481271863816  
Prediction\_local [279.49715345]  
Right: 279.6475186812419  
Intercept 279.76155629067324  
Prediction\_local [279.56906512]  
Right: 279.64149130804293  
Intercept 279.6411745734793  
Prediction\_local [279.42904304]  
Right: 279.52486742414834  
Intercept 279.38230895324796  
Prediction\_local [280.09559909]  
Right: 280.21771234234046  
Intercept 279.7622351959747  
Prediction\_local [279.13977422]  
Right: 278.8493549759595  
Intercept 279.4906592960395  
Prediction\_local [280.03136047]  
Right: 280.3476623773903  
Intercept 279.62437241691003  
Prediction\_local [279.41070887]  
Right: 279.55961470948574  
Intercept 279.49228523076545  
Prediction\_local [279.65291836]  
Right: 279.83154546119636  
Intercept 279.4332625880936  
Prediction\_local [280.07074472]  
Right: 280.26896874525306  
Intercept 279.46620233101436  
Prediction\_local [279.76112888]  
Right: 280.0145283426927  
Intercept 279.7415025212  
Prediction\_local [279.57667048]  
Right: 279.63441736075214  
Intercept 279.4095423608159  
Prediction\_local [279.42985229]  
Right: 279.5501177758108  
Intercept 279.74071219366397  
Prediction\_local [279.35183735]  
Right: 279.33099465304815  
Intercept 279.3840516521685  
Prediction\_local [279.6847837]  
Right: 279.7671125974717  
Intercept 279.72909469320024  
Prediction\_local [279.05569154]  
Right: 278.7430509859108  
Intercept 279.65852652611045  
Prediction\_local [279.24785596]  
Right: 279.03541655928916  
Intercept 279.3625038749205  
Prediction\_local [279.78803373]  
Right: 280.01184461262346  
Intercept 279.71883057858247  
Prediction\_local [279.32232347]  
Right: 279.263527880082  
Intercept 279.39490868920933  
Prediction\_local [280.01439334]  
Right: 280.22627495054803  
Intercept 279.43272371574244  
Prediction\_local [280.10671376]  
Right: 280.3356316957793  
Intercept 279.40671321052895  
Prediction\_local [280.11808764]  
Right: 280.345974444389  
Intercept 279.58612174027013  
Prediction\_local [279.43973705]  
Right: 279.4725084512741  
Intercept 279.627233519819  
Prediction\_local [279.3843187]  
Right: 278.6510156011505

exp

<lime.explanation.Explanation at 0x25d8d8a4880>

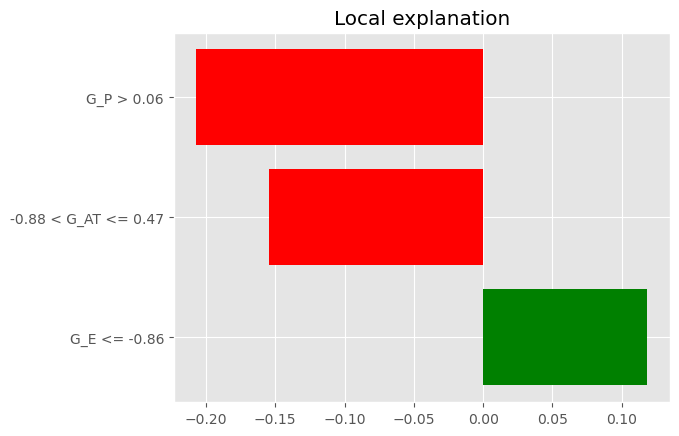
### Printing the explanation  
# print('Ínstance:', i)

#### Printing the prediction  
# print('Prediction:', y\_R\_pred\_test[i])

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
with plt.style.context('ggplot'):  
 fig = plt.figure(figsize = (8,5))  
 plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
 plt.yticks(range(len(model.coef\_)), X\_G\_test.columns);  
 plt.title('Weights')



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_P > 0.06', -0.20692449502909413),  
 ('-0.88 < G\_AT <= 0.47', -0.15442420359163284),  
 ('G\_E <= -0.86', 0.11843387966225495)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 0.20692449502909413),  
 (1, 0.15442420359163284),  
 (0, -0.11843387966225495)],  
 1: [(2, -0.20692449502909413),  
 (1, -0.15442420359163284),  
 (0, 0.11843387966225495)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.3843187]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 278.6510156011505

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_GG\_test.html')

##################### END ####################################

#######################################################################################################

#######################################################################################################

#######################################################################################################

##################### SUPPORT VECTOR REGRESSION ###################

from sklearn.svm import SVR  
seed = 7

############ MODEL 1: RS Features (X\_R) and RS Target (LL\_R)###############

########### On the whole dataset  
model = SVR()  
svr\_param\_grid = {'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],  
 'epsilon' : [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'C' : [1, 30, 60, 90, 120, 150, 180]  
 }  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr1 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
svr1.fit(X\_R, y\_R)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [1, 30, 60, 90, 120, 150, 180],  
 'epsilon': [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'kernel': ['linear', 'rbf', 'poly', 'sigmoid']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The svr best parameters are:', svr1.best\_params\_)  
print('The svr best score is:', svr1.best\_score\_)

The svr best parameters are: {'C': 60, 'epsilon': 0.001, 'kernel': 'poly'}  
The svr best score is: -0.10500453922121007

# The svr best parameters are: {'C': 60, 'epsilon': 0.001, 'kernel': 'poly'}  
# The svr best score is: -0.10500453922121007

# Evaluation of the performance of the model on the whole dataset:  
svr\_y\_pred = svr1.predict(X\_R)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_R, svr\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_R, svr\_y\_pred))  
print('R2 Score:', r2\_score(y\_R, svr\_y\_pred))

Mean Absolute Error: 0.23387264316141015  
Mean Squared Error: 0.09465960233585312  
R2 Score: 0.536162694594717

# Mean Absolute Error: 0.233524467695166  
# Mean Squared Error: 0.09521109486663822  
# R2 Score: 0.5334603505839857

################ LIME ################

###### Initializing the explainer on the whole features dataset  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the full datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, svr1.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 281.0734227128305  
Prediction\_local [281.5377846]  
Right: 281.57868409022296  
Intercept 281.17664867482927  
Prediction\_local [281.45849301]  
Right: 281.4693043248706  
Intercept 281.0411161076652  
Prediction\_local [281.01807182]  
Right: 281.2894095409258  
Intercept 281.2052297269961  
Prediction\_local [281.05485254]  
Right: 281.2000767561815  
Intercept 281.12827321337346  
Prediction\_local [281.15697589]  
Right: 281.2885932593709  
Intercept 281.1441431581944  
Prediction\_local [281.18567643]  
Right: 281.08821930679824  
Intercept 281.11988304317947  
Prediction\_local [280.41267083]  
Right: 281.334926860001  
Intercept 281.21119156645824  
Prediction\_local [281.21005477]  
Right: 281.5377377436477  
Intercept 280.81289867473424  
Prediction\_local [281.71783817]  
Right: 281.6349774607522  
Intercept 280.7160668903846  
Prediction\_local [281.87359088]  
Right: 281.8603421921376  
Intercept 281.13640143224717  
Prediction\_local [281.45470947]  
Right: 281.5747446522746  
Intercept 280.9188690744233  
Prediction\_local [281.68173265]  
Right: 281.65389505804075  
Intercept 280.9165924296161  
Prediction\_local [281.7109108]  
Right: 281.63473657436  
Intercept 281.19859860777393  
Prediction\_local [281.4412173]  
Right: 281.5561918048496  
Intercept 280.6932562788027  
Prediction\_local [281.3380984]  
Right: 281.32693932027325  
Intercept 281.014202260857  
Prediction\_local [280.92629715]  
Right: 281.22288170827005  
Intercept 281.04099570655205  
Prediction\_local [280.99373321]  
Right: 280.89637167526524  
Intercept 281.1611568137076  
Prediction\_local [281.24921643]  
Right: 280.86942046181775  
Intercept 281.27367165798137  
Prediction\_local [280.07449421]  
Right: 280.8411288579381  
Intercept 281.2512272088386  
Prediction\_local [280.86788948]  
Right: 281.4745208969314  
Intercept 281.07364736319266  
Prediction\_local [281.57369012]  
Right: 281.64099022383573  
Intercept 281.12393526592035  
Prediction\_local [281.45986245]  
Right: 281.82564162221695  
Intercept 281.0677183266791  
Prediction\_local [281.30400059]  
Right: 281.6202131712683  
Intercept 281.01968210673846  
Prediction\_local [281.58255394]  
Right: 281.65625179861627  
Intercept 280.87510531339944  
Prediction\_local [281.79837429]  
Right: 281.7665456834273  
Intercept 280.9178338370725  
Prediction\_local [281.25801265]  
Right: 281.4771463070312  
Intercept 281.1161456621762  
Prediction\_local [281.18084113]  
Right: 281.33390354855425  
Intercept 281.1402794282863  
Prediction\_local [281.02035341]  
Right: 281.32143006283064  
Intercept 281.2832981216312  
Prediction\_local [280.7027986]  
Right: 281.1852914364296  
Intercept 281.0052629401084  
Prediction\_local [281.54658296]  
Right: 280.92409665756793  
Intercept 281.41183610708873  
Prediction\_local [280.76827973]  
Right: 281.15158635386257  
Intercept 281.29529992143375  
Prediction\_local [280.6506653]  
Right: 281.1108811779426  
Intercept 281.35289314761803  
Prediction\_local [280.76588768]  
Right: 281.41827788951343  
Intercept 281.03278545994095  
Prediction\_local [280.99635899]  
Right: 281.5414918448603  
Intercept 281.0892770613303  
Prediction\_local [281.50284787]  
Right: 281.6141984002192  
Intercept 281.0869794612781  
Prediction\_local [281.32106856]  
Right: 281.79864318487114  
Intercept 281.21786447411125  
Prediction\_local [281.39972286]  
Right: 281.84878493715166  
Intercept 281.1586501027328  
Prediction\_local [281.23030088]  
Right: 281.7127434524568  
Intercept 280.9628934166461  
Prediction\_local [281.339494]  
Right: 281.2235675619487  
Intercept 280.93197690851605  
Prediction\_local [281.1463222]  
Right: 281.14218664604545  
Intercept 281.1927358379887  
Prediction\_local [281.08739639]  
Right: 281.0423426315279  
Intercept 281.45209964858566  
Prediction\_local [280.04210696]  
Right: 280.79519663595835  
Intercept 281.28655950222776  
Prediction\_local [280.45279981]  
Right: 280.9060141521966  
Intercept 281.46406929412024  
Prediction\_local [280.38764324]  
Right: 281.1872057684389  
Intercept 281.2064902614598  
Prediction\_local [280.65587714]  
Right: 281.4235866818299  
Intercept 280.99822908855475  
Prediction\_local [281.12424784]  
Right: 281.5091353137224  
Intercept 281.0985452033452  
Prediction\_local [281.39274297]  
Right: 281.53601285701586  
Intercept 280.8282607191126  
Prediction\_local [281.76180335]  
Right: 281.7278248615603  
Intercept 281.0678789733466  
Prediction\_local [281.61720642]  
Right: 281.7254073493983  
Intercept 280.94897604660275  
Prediction\_local [281.40596103]  
Right: 281.76104000815025  
Intercept 281.0189998514042  
Prediction\_local [281.3408694]  
Right: 281.4947472031592  
Intercept 281.1900330016301  
Prediction\_local [280.65596281]  
Right: 281.1254329191099  
Intercept 281.11573680314063  
Prediction\_local [280.91728595]  
Right: 281.0984581163664  
Intercept 281.4341355905721  
Prediction\_local [279.92170419]  
Right: 281.0931349280015  
Intercept 281.4015235216239  
Prediction\_local [280.25635099]  
Right: 280.9446574939542  
Intercept 281.3834957270642  
Prediction\_local [280.52010852]  
Right: 280.93239087909944  
Intercept 281.5991226819166  
Prediction\_local [279.69509407]  
Right: 281.02377137332513  
Intercept 281.1048769211505  
Prediction\_local [280.98798046]  
Right: 281.42666960987685  
Intercept 281.15359514224696  
Prediction\_local [281.36453376]  
Right: 281.5248660933738  
Intercept 281.0192120013706  
Prediction\_local [281.54915247]  
Right: 281.6433052078293  
Intercept 280.92740700254757  
Prediction\_local [281.57967157]  
Right: 281.7908516422828  
Intercept 281.00345057587936  
Prediction\_local [281.56502836]  
Right: 281.47925182608043  
Intercept 281.2189459466141  
Prediction\_local [280.91626636]  
Right: 281.12305064986316  
Intercept 281.253205024812  
Prediction\_local [281.10665019]  
Right: 281.1336349195234  
Intercept 281.0367867416108  
Prediction\_local [281.38282832]  
Right: 280.8191090060402  
Intercept 281.3647242028795  
Prediction\_local [280.19446648]  
Right: 280.36056851991736  
Intercept 281.3247929744433  
Prediction\_local [280.26888686]  
Right: 280.8905079105209  
Intercept 281.2254697887931  
Prediction\_local [280.7237872]  
Right: 281.3524970282612  
Intercept 281.26966706175006  
Prediction\_local [280.24223631]  
Right: 281.340990132174  
Intercept 281.0400420041567  
Prediction\_local [281.41280754]  
Right: 281.6644002807598  
Intercept 280.9138036779047  
Prediction\_local [281.58881327]  
Right: 281.51345610345606  
Intercept 280.9455296725366  
Prediction\_local [281.58510787]  
Right: 281.75495340306674  
Intercept 280.99804343177493  
Prediction\_local [281.48116701]  
Right: 281.73682063718184  
Intercept 280.9491954887148  
Prediction\_local [281.67334345]  
Right: 281.7064898039314  
Intercept 280.8086991630501  
Prediction\_local [281.12929915]  
Right: 281.369091104444  
Intercept 281.2773847052843  
Prediction\_local [280.67624286]  
Right: 281.07664468448775  
Intercept 281.113893881614  
Prediction\_local [280.65514489]  
Right: 280.0483241541972  
Intercept 281.4019645211827  
Prediction\_local [280.42331357]  
Right: 280.96009080740794  
Intercept 281.316228482387  
Prediction\_local [280.22896741]  
Right: 281.0384928489168  
Intercept 281.1442022383848  
Prediction\_local [280.59307107]  
Right: 281.3287880489966  
Intercept 281.28468549744616  
Prediction\_local [280.60299228]  
Right: 281.2706235324613  
Intercept 281.3766717814181  
Prediction\_local [280.76344386]  
Right: 281.46808414397196  
Intercept 280.90555396896696  
Prediction\_local [281.67345737]  
Right: 281.6336348214927  
Intercept 280.9649616879903  
Prediction\_local [281.5468848]  
Right: 281.78330536943133  
Intercept 280.8956570349098  
Prediction\_local [281.51149012]  
Right: 281.8231463219534  
Intercept 281.07651173217135  
Prediction\_local [281.4368]  
Right: 281.74645197541975  
Intercept 281.1332162067855  
Prediction\_local [281.20559373]  
Right: 281.42165472967025  
Intercept 281.1180071196662  
Prediction\_local [281.00861661]  
Right: 281.16139826447943  
Intercept 281.0582586679441  
Prediction\_local [281.04661411]  
Right: 281.2154804768597  
Intercept 281.08621687449204  
Prediction\_local [281.2602276]  
Right: 281.190494483504  
Intercept 281.26408369880875  
Prediction\_local [280.66774145]  
Right: 281.16532397086354  
Intercept 281.32966359792437  
Prediction\_local [280.82734018]  
Right: 281.49950151064223  
Intercept 281.3170519169576  
Prediction\_local [280.56221143]  
Right: 281.2584138577609  
Intercept 281.1601818367959  
Prediction\_local [281.41838493]  
Right: 282.0307209610055  
Intercept 281.1669194494572  
Prediction\_local [281.56440186]  
Right: 281.6120356661414  
Intercept 280.79178884078  
Prediction\_local [281.8827963]  
Right: 281.675615551948  
Intercept 281.20032414923327  
Prediction\_local [281.59508863]  
Right: 281.72073698021313  
Intercept 281.000713728187  
Prediction\_local [281.74039131]  
Right: 281.7372347442012  
Intercept 280.7168728872958  
Prediction\_local [281.80729036]  
Right: 281.78579875856235  
Intercept 281.1619449062103  
Prediction\_local [281.08794792]  
Right: 281.3949499148221  
Intercept 281.0397027820061  
Prediction\_local [281.69348136]  
Right: 281.2920321810511  
Intercept 280.81015796903336  
Prediction\_local [281.59373056]  
Right: 281.6366096684637  
Intercept 281.180170793037  
Prediction\_local [281.21907006]  
Right: 280.92911873370423  
Intercept 280.86376805782476  
Prediction\_local [281.54037695]  
Right: 281.7951990912357  
Intercept 281.0050055938695  
Prediction\_local [281.75424181]  
Right: 281.8211554951875  
Intercept 281.06735916339017  
Prediction\_local [281.49595923]  
Right: 281.991089172342  
Intercept 280.8487036025317  
Prediction\_local [281.43559597]  
Right: 281.5225579781082  
Intercept 280.7514351295835  
Prediction\_local [281.56659358]  
Right: 281.7800403440413

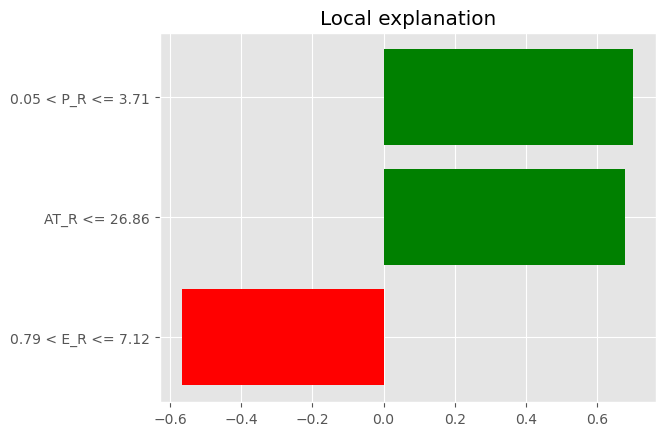
exp

<lime.explanation.Explanation at 0x24a265d9250>

##### Showing the explanable table  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the svr  
#with plt.style.context('ggplot'):  
# fig = plt.figure(figsize = (8,5))  
# plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
# plt.yticks(range(len(model.coef\_)), X\_R.columns);  
# plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('0.05 < P\_R <= 3.71', 0.7018436614639701),  
 ('AT\_R <= 26.86', 0.6794077610427687),  
 ('0.79 < E\_R <= 7.12', -0.5660929751186654)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.7018436614639701),  
 (1, -0.6794077610427687),  
 (0, 0.5660929751186654)],  
 1: [(2, 0.7018436614639701),  
 (1, 0.6794077610427687),  
 (0, -0.5660929751186654)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.56659358]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.7800403440413

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr1RR\_Whole.html')

####################################################################################################

########### Fitting the model on the training dataset using the best parameters  
model = SVR()  
svr\_param\_grid = {'kernel': ['poly'],  
 'epsilon' : [0.001],  
 'C' : [60]  
 }  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr1 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1 )  
svr1.fit(X\_R\_train, y\_R\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [60], 'epsilon': [0.001], 'kernel': ['poly']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The svr best score is:', svr1.best\_score\_)

The svr best score is: -0.1561231841307928

# Evaluation of the performance of the model on the training dataset:  
svr\_y\_predtr = svr1.predict(X\_R\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_R\_train, svr\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_R\_train, svr\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_R\_train, svr\_y\_predtr))

The training MAE is: 0.2882061496224825  
The training MSE is: 0.13954093157780578  
The training R2 Score is: 0.3394913134536164

# Evaluation of the performance of the model on the testing dataset:  
svr\_y\_predts = svr1.predict(X\_R\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_R\_test, svr\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_R\_test, svr\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_R\_test, svr\_y\_predts))

The testing MAE is: 0.32534706687286  
The testing MSE is: 0.16635487439795374  
The testing R2 Score is: -0.14534414747673785

########### Cross-validation ###########################

############# On the training dataset  
   
score\_svr1tr = cross\_val\_score(svr1, X\_R\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_svr1tr

array([-0.18448063, -0.04781955, -0.17587177, -0.20750256, -0.25133934,  
 -0.17966564, -0.06540751, -0.14680268, -0.25602054, -0.14463057])

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_svr1tr)))

The mean training CV score is: 0.16595408044017113

############# On the testing dataset   
score\_svr1ts = cross\_val\_score(svr1, X\_R\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

score\_svr1ts

array([-0.03849409, -0.53143449, -0.06550355, -0.11605462, -0.26309246,  
 -0.11764025, -1.44143603, -0.04362415, -0.17237967, -0.0617194 ])

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_svr1ts)))

The mean testing CV score is: 0.28513787058805884

################ LIME ################

###### Initializing the explainer on the training dataset  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, svr1.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.661327577202  
Prediction\_local [282.12132721]  
Right: 281.33432003876135  
Intercept 281.9707403221628  
Prediction\_local [280.3726798]  
Right: 281.27451447408146  
Intercept 281.56087517820873  
Prediction\_local [281.36511826]  
Right: 281.5407854850061  
Intercept 281.59080111326784  
Prediction\_local [281.83312566]  
Right: 280.7402301327999  
Intercept 281.51284791703335  
Prediction\_local [282.54558953]  
Right: 281.1105070122986  
Intercept 282.03218550891177  
Prediction\_local [281.11077322]  
Right: 281.4548750617627  
Intercept 282.01631204747514  
Prediction\_local [282.01731527]  
Right: 280.89724702241085  
Intercept 281.7199405671471  
Prediction\_local [281.28803612]  
Right: 281.3104740187809  
Intercept 281.29314999363345  
Prediction\_local [281.73867901]  
Right: 281.09965572981633  
Intercept 281.6117692058774  
Prediction\_local [281.61202329]  
Right: 281.04392792344726  
Intercept 282.10936044047634  
Prediction\_local [281.46588743]  
Right: 281.42208431291823  
Intercept 281.7284221010573  
Prediction\_local [281.46458228]  
Right: 280.94578562660644  
Intercept 281.7508844560775  
Prediction\_local [281.75804682]  
Right: 281.38599611016434  
Intercept 281.5255475280695  
Prediction\_local [281.79455929]  
Right: 281.5602917109934  
Intercept 281.37088679931895  
Prediction\_local [282.02300729]  
Right: 281.73417253807713  
Intercept 281.08621152827396  
Prediction\_local [283.73355305]  
Right: 280.9590148851585  
Intercept 281.75580778174583  
Prediction\_local [282.21643425]  
Right: 281.45792466303527  
Intercept 280.9606464203213  
Prediction\_local [284.31041015]  
Right: 281.45526909950866  
Intercept 281.8834951570367  
Prediction\_local [281.20769937]  
Right: 281.3976902631807  
Intercept 281.3745139836687  
Prediction\_local [281.45471826]  
Right: 281.3035205255122  
Intercept 281.1779532819384  
Prediction\_local [283.20070224]  
Right: 281.3785408573148  
Intercept 281.5579805578554  
Prediction\_local [282.66011182]  
Right: 281.75222522955164  
Intercept 281.5609749968041  
Prediction\_local [281.90357127]  
Right: 281.2917227332716  
Intercept 282.44633464330917  
Prediction\_local [280.67803448]  
Right: 281.11692216737544  
Intercept 281.49639538054026  
Prediction\_local [281.22226682]  
Right: 281.2898917251238  
Intercept 281.8782303777557  
Prediction\_local [282.25671535]  
Right: 281.68150953068806  
Intercept 281.86050854449377  
Prediction\_local [281.72257981]  
Right: 281.40199058862976  
Intercept 281.84590332644774  
Prediction\_local [281.67302148]  
Right: 281.69322778605556  
Intercept 281.7444556865557  
Prediction\_local [281.76646877]  
Right: 280.8537507065821  
Intercept 281.40548620644944  
Prediction\_local [280.90992964]  
Right: 281.42533837186113  
Intercept 282.02886024029425  
Prediction\_local [281.2357311]  
Right: 280.95136445713086  
Intercept 281.75060536915004  
Prediction\_local [281.52835819]  
Right: 281.4417742959236  
Intercept 281.5039575119529  
Prediction\_local [281.54001466]  
Right: 282.2295061975118  
Intercept 281.58569217641934  
Prediction\_local [282.01961548]  
Right: 281.449098723323  
Intercept 281.1337739421082  
Prediction\_local [282.42951708]  
Right: 281.6626052933071  
Intercept 281.66429913801556  
Prediction\_local [281.17238573]  
Right: 281.1770937635577  
Intercept 281.31058468807163  
Prediction\_local [281.51811205]  
Right: 281.6597556381604  
Intercept 281.6077949896586  
Prediction\_local [282.83340661]  
Right: 281.37279875958757  
Intercept 281.47281345714464  
Prediction\_local [281.07659744]  
Right: 280.89736630010736  
Intercept 282.31945931589354  
Prediction\_local [280.93541856]  
Right: 281.2736305324948  
Intercept 282.31665348820536  
Prediction\_local [281.302599]  
Right: 281.4163700925374  
Intercept 280.59718669747866  
Prediction\_local [284.34351138]  
Right: 281.6912691664751  
Intercept 282.1366575189386  
Prediction\_local [281.36896149]  
Right: 281.32476915712937  
Intercept 282.73677153725646  
Prediction\_local [281.4168905]  
Right: 281.4332693149881  
Intercept 282.5211517163742  
Prediction\_local [280.6764973]  
Right: 281.21724532098665  
Intercept 281.5314798079969  
Prediction\_local [281.55201954]  
Right: 281.3860813521244  
Intercept 281.40385069105434  
Prediction\_local [281.36318503]  
Right: 281.43378913958924  
Intercept 281.5674755272763  
Prediction\_local [281.03294363]  
Right: 281.4455053361594  
Intercept 282.12306871973726  
Prediction\_local [280.54209134]  
Right: 281.4801531667897  
Intercept 281.6622128086678  
Prediction\_local [281.60640228]  
Right: 280.9313023551479  
Intercept 282.74867095320747  
Prediction\_local [281.27976159]  
Right: 281.2174652058084  
Intercept 281.7137235826537  
Prediction\_local [280.2605587]  
Right: 281.54635165356717  
Intercept 281.22258146000786  
Prediction\_local [281.40391635]  
Right: 282.059352209642  
Intercept 281.6255382544501  
Prediction\_local [281.18813979]  
Right: 280.81928639866817  
Intercept 281.64566258234527  
Prediction\_local [281.90471004]  
Right: 281.5430248794017  
Intercept 282.23853341958926  
Prediction\_local [281.30390392]  
Right: 281.4211771626265  
Intercept 281.9381215396813  
Prediction\_local [281.63019956]  
Right: 281.31721852965336  
Intercept 281.3360982387306  
Prediction\_local [281.03869578]  
Right: 281.253811234365  
Intercept 281.5645807671341  
Prediction\_local [281.76917616]  
Right: 281.5199585731088  
Intercept 281.908332415851  
Prediction\_local [281.37620482]  
Right: 281.3117756803691  
Intercept 282.25328225751474  
Prediction\_local [280.75666778]  
Right: 280.83895559775647  
Intercept 280.8238473178993  
Prediction\_local [282.05751994]  
Right: 281.17086251743245  
Intercept 281.95281769489804  
Prediction\_local [280.57674223]  
Right: 281.3605243069062  
Intercept 281.6565051654585  
Prediction\_local [282.17104569]  
Right: 281.88760049009716  
Intercept 282.4482885077655  
Prediction\_local [280.66375078]  
Right: 281.4272874980721  
Intercept 281.6040212190276  
Prediction\_local [282.29331954]  
Right: 281.740520341264  
Intercept 281.87524041604894  
Prediction\_local [282.48457578]  
Right: 281.7876405493573  
Intercept 282.57577460812854  
Prediction\_local [279.6262792]  
Right: 281.45457620875675  
Intercept 281.29722134818115  
Prediction\_local [282.5100548]  
Right: 281.20175461451475  
Intercept 282.15360767828287  
Prediction\_local [279.46948335]  
Right: 281.09420226729645  
Intercept 282.10069760342844  
Prediction\_local [280.63621401]  
Right: 281.1494595128845  
Intercept 282.009382204144  
Prediction\_local [281.57123513]  
Right: 281.4381298653391  
Intercept 281.97279018502627  
Prediction\_local [281.33863135]  
Right: 281.3143543304132  
Intercept 281.2275875731062  
Prediction\_local [280.95852631]  
Right: 280.9196051139568  
Intercept 281.8571061658064  
Prediction\_local [280.88660158]  
Right: 281.2214485310427  
Intercept 281.5433817161958  
Prediction\_local [280.93223224]  
Right: 281.54396138377064  
Intercept 282.1821641638715  
Prediction\_local [282.09753282]  
Right: 281.5458100523992  
Intercept 282.236211546007  
Prediction\_local [281.09636815]  
Right: 281.36966342016024  
Intercept 281.012449119245  
Prediction\_local [281.97410804]  
Right: 281.3518370667197  
Intercept 281.7176278142962  
Prediction\_local [282.30068389]  
Right: 281.6588391434679  
Intercept 281.9434823002068  
Prediction\_local [283.00019242]  
Right: 281.65848908340286

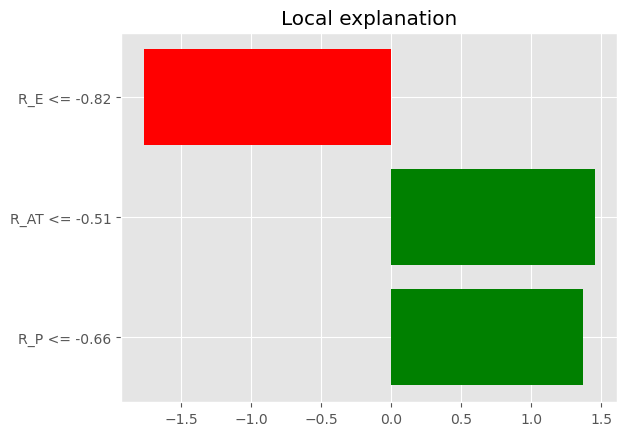
exp

<lime.explanation.Explanation at 0x25d8e750220>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
#with plt.style.context('ggplot'):  
# fig = plt.figure(figsize = (8,5))  
# plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
# plt.yticks(range(len(model.coef\_)), X\_R\_train.columns);  
# plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_E <= -0.82', -1.7674447553166495),  
 ('R\_AT <= -0.51', 1.4555007652036436),  
 ('R\_P <= -0.66', 1.3686541073342904)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(0, 1.7674447553166495),  
 (1, -1.4555007652036436),  
 (2, -1.3686541073342904)],  
 1: [(0, -1.7674447553166495),  
 (1, 1.4555007652036436),  
 (2, 1.3686541073342904)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [283.00019242]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.65848908340286

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr1RR\_train.html')

################################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, svr1.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 280.49593376151336  
Prediction\_local [280.67889784]  
Right: 281.4184222652878  
Intercept 282.2713254943053  
Prediction\_local [281.0356939]  
Right: 281.48986970710774  
Intercept 281.87915033147584  
Prediction\_local [281.13097412]  
Right: 280.99286571200616  
Intercept 280.8657228768624  
Prediction\_local [284.1321442]  
Right: 282.46950544474845  
Intercept 280.492100653645  
Prediction\_local [281.24686293]  
Right: 281.3127907281569  
Intercept 282.2069381763772  
Prediction\_local [281.38204595]  
Right: 281.98933798883326  
Intercept 280.67577671215247  
Prediction\_local [284.00876648]  
Right: 281.67556849225775  
Intercept 280.48282115472267  
Prediction\_local [282.16641718]  
Right: 281.72885184543895  
Intercept 282.14974818588234  
Prediction\_local [282.87326868]  
Right: 281.1593863926584  
Intercept 282.0988102047981  
Prediction\_local [283.26844679]  
Right: 281.8139628074341  
Intercept 281.3156534250531  
Prediction\_local [280.39298707]  
Right: 281.06621317002623  
Intercept 281.51386921767715  
Prediction\_local [281.69895825]  
Right: 281.70747270083456  
Intercept 281.88353260844445  
Prediction\_local [283.65164843]  
Right: 281.73440872772153  
Intercept 281.25291801971656  
Prediction\_local [281.48204122]  
Right: 281.41288597932834  
Intercept 282.041459808946  
Prediction\_local [281.15501526]  
Right: 281.0973960432428  
Intercept 282.1079682650329  
Prediction\_local [281.2152034]  
Right: 281.0514852376585  
Intercept 281.6848653292278  
Prediction\_local [282.64846927]  
Right: 282.1264491000626  
Intercept 281.9505486664083  
Prediction\_local [281.31272391]  
Right: 281.64642443082084  
Intercept 281.56767621326514  
Prediction\_local [282.89914088]  
Right: 280.99929153743824  
Intercept 281.5961646672588  
Prediction\_local [281.37753059]  
Right: 280.68208083438253  
Intercept 282.0498224216625  
Prediction\_local [282.34638924]  
Right: 281.42692429332965  
Intercept 281.537648222085  
Prediction\_local [282.02984046]  
Right: 280.6398021927671  
Intercept 281.1631668982329  
Prediction\_local [281.68358632]  
Right: 281.92512484726694  
Intercept 282.4950220055367  
Prediction\_local [281.57471447]  
Right: 281.7535840287336  
Intercept 281.4513571320317  
Prediction\_local [281.82907218]  
Right: 281.57314219271166  
Intercept 282.93850375708774  
Prediction\_local [280.30676002]  
Right: 281.0039615167366  
Intercept 281.6287080472127  
Prediction\_local [281.1995868]  
Right: 280.9687052857144

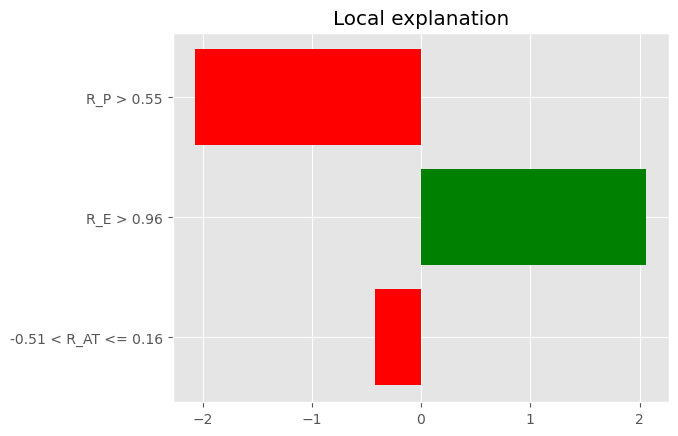
exp

<lime.explanation.Explanation at 0x25d8e24a790>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
#with plt.style.context('ggplot'):  
# fig = plt.figure(figsize = (8,5))  
# plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])## plt.yticks(range(len(model.coef\_)), X\_R\_test.columns);  
# plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P > 0.55', -2.0653737966711696),  
 ('R\_E > 0.96', 2.0619394389485253),  
 ('-0.51 < R\_AT <= 0.16', -0.42568689179553953)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 2.0653737966711696),  
 (0, -2.0619394389485253),  
 (1, 0.42568689179553953)],  
 1: [(2, -2.0653737966711696),  
 (0, 2.0619394389485253),  
 (1, -0.42568689179553953)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.1995868]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.9687052857144

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr1RR\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

############ MODEL 2: GT Features (X\_G) and RS Target (LL\_R) ###############

########### On the whole dataset  
model = SVR()  
svr\_param\_grid = {'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],  
 'epsilon' : [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'C' : [1, 30, 60, 90, 120,150,180]  
 }  
  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr2 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
svr2.fit(X\_G, y\_R)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [1, 30, 60, 90, 120, 150, 180],  
 'epsilon': [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'kernel': ['linear', 'rbf', 'poly', 'sigmoid']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The svr best parameters are:', svr2.best\_params\_)  
print('The svr best score is:', svr2.best\_score\_)

The svr best parameters are: {'C': 150, 'epsilon': 0.1, 'kernel': 'rbf'}  
The svr best score is: -0.10396687742412233

#The svr best parameters are: {'C': 150, 'epsilon': 0.1, 'kernel': 'rbf'}  
#The svr best score is: -0.10396687742412233

# Evaluation of the performance of the svr regression model on the whole dataset:  
svr\_y\_pred = svr2.predict(X\_G)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_R, svr\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_R, svr\_y\_pred))  
print('R2 Score:', r2\_score(y\_R, svr\_y\_pred))

Mean Absolute Error: 0.19276205999392143  
Mean Squared Error: 0.06122029796242759  
R2 Score: 0.7000171420301262

#Mean Absolute Error: 0.19276205999392143  
#Mean Squared Error: 0.06122029796242759  
#R2 Score: 0.7000171420301262

#####################LIME ##############################

###### Initializing the explainer on the whole datset  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the whole XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, svr2.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.473985261616  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161624  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.47398526161606  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616161  
Prediction\_local [282.47398526]  
Right: 282.4739852616161  
Intercept 282.4739852616162  
Prediction\_local [282.47398526]  
Right: 282.4739852616161

exp

<lime.explanation.Explanation at 0x24a27562610>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('AT\_G <= 25.38', 0.7055089896952325),  
 ('P\_G <= 0.00', 0.19234621768688218),  
 ('9.70 < E\_G <= 13.62', -0.18461774873475464)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.7055089896952325),  
 (2, -0.19234621768688218),  
 (0, 0.18461774873475464)],  
 1: [(1, 0.7055089896952325),  
 (2, 0.19234621768688218),  
 (0, -0.18461774873475464)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [282.04700562]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.89799954228323

# Saving fe# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr2GR\_Whole.html')

##########################################################################################

########### Fitting the model on the training dataset using the best parameters  
model = SVR()  
svr\_param\_grid = {'kernel': ['rbf'],  
 'epsilon' : [0.1],  
 'C' : [150]  
 }  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr2 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1 )  
svr2.fit(X\_G\_train, y\_R\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [150], 'epsilon': [0.1], 'kernel': ['rbf']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The svr best score is:', svr2.best\_score\_)

The svr best score is: -0.09656597473446349

# Evaluation of the performance of the model on the training dataset:  
svr\_y\_predtr = svr2.predict(X\_G\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, svr\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, svr\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, svr\_y\_predtr))

The training MAE is: 1.6508441090368384  
The training MSE is: 2.821356526366474  
The training R2 Score is: -8.253500193404188

#The training MAE is: 1.6508441090368384  
#The training MSE is: 2.821356526366474  
#The training R2 Score is: -8.253500193404188

# Evaluation of the performance of the model on the testing dataset:  
svr\_y\_predts = svr2.predict(X\_G\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, svr\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, svr\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, svr\_y\_predts))

The testing MAE is: 1.94040020352059  
The testing MSE is: 5.133279223578421  
The testing R2 Score is: -17.718739515497887

#The testing MAE is: 1.94040020352059  
#The testing MSE is: 5.133279223578421  
#The testing R2 Score is: -17.718739515497887

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_svr2tr = cross\_val\_score(svr2, X\_G\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_svr2tr)))

The mean training CV score is: 0.09112277797745014

############# On the testing dataset   
score\_svr2ts = cross\_val\_score(svr2, X\_G\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_svr2ts)))

The mean testing CV score is: 0.11680173809226149

#####################LIME ##############################

###### Initializing the explainer on the training dataset  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, svr2.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.8352878344397  
Prediction\_local [281.44299135]  
Right: 281.2895442629629  
Intercept 281.66011179579726  
Prediction\_local [281.44516364]  
Right: 281.0615114074113  
Intercept 281.3103875842577  
Prediction\_local [282.03931686]  
Right: 281.9388693585559  
Intercept 281.6017531392219  
Prediction\_local [281.48823257]  
Right: 281.1069563081011  
Intercept 281.54589990115664  
Prediction\_local [281.36414509]  
Right: 281.21006705937714  
Intercept 281.41154948304586  
Prediction\_local [281.90145339]  
Right: 282.06452571676715  
Intercept 281.6859671185196  
Prediction\_local [281.61073999]  
Right: 281.04637198927304  
Intercept 281.92895786802126  
Prediction\_local [280.90585393]  
Right: 280.9081245375202  
Intercept 281.81258105926327  
Prediction\_local [280.84824891]  
Right: 280.8697586201004  
Intercept 281.4366031425389  
Prediction\_local [281.54118424]  
Right: 280.9698741973777  
Intercept 281.5726134484527  
Prediction\_local [281.32168824]  
Right: 281.86963515150546  
Intercept 281.6026379228111  
Prediction\_local [281.43198213]  
Right: 281.06764963447444  
Intercept 281.76789840974607  
Prediction\_local [281.51285381]  
Right: 281.3854646725425  
Intercept 281.66233984464816  
Prediction\_local [281.63842092]  
Right: 281.3604151653463  
Intercept 281.6517007340834  
Prediction\_local [281.95898386]  
Right: 281.8614106982043  
Intercept 281.5530948360544  
Prediction\_local [281.12537358]  
Right: 280.8595932067714  
Intercept 281.7314125629354  
Prediction\_local [281.55125928]  
Right: 281.73841844672756  
Intercept 281.68779199071867  
Prediction\_local [281.55215234]  
Right: 281.700052119423  
Intercept 281.76504184592375  
Prediction\_local [281.13834022]  
Right: 280.6475532746684  
Intercept 281.59173957187505  
Prediction\_local [281.46746903]  
Right: 280.82982987880735  
Intercept 281.5978257825528  
Prediction\_local [281.5571893]  
Right: 281.2001771559531  
Intercept 281.50710152737054  
Prediction\_local [282.06810726]  
Right: 281.78403101184045  
Intercept 281.4912476573772  
Prediction\_local [281.58507269]  
Right: 281.37124087147464  
Intercept 281.6568535253145  
Prediction\_local [281.5394932]  
Right: 281.0627606458942  
Intercept 281.639499521889  
Prediction\_local [281.59575813]  
Right: 281.2084524735408  
Intercept 281.4355731723094  
Prediction\_local [281.97089962]  
Right: 281.74895503747405  
Intercept 281.61423847438033  
Prediction\_local [281.19539671]  
Right: 280.9257660914113  
Intercept 281.5474175946781  
Prediction\_local [281.96209131]  
Right: 281.55006456228665  
Intercept 281.4670493100166  
Prediction\_local [281.42292493]  
Right: 281.0402525008121  
Intercept 281.4935597927652  
Prediction\_local [281.89622133]  
Right: 281.4887393730404  
Intercept 281.59359515845523  
Prediction\_local [281.18841847]  
Right: 280.6502275291367  
Intercept 281.56336257201775  
Prediction\_local [282.01498989]  
Right: 282.0404461018092  
Intercept 281.46165601099426  
Prediction\_local [281.8843391]  
Right: 281.37060151978335  
Intercept 281.79498977996474  
Prediction\_local [281.17167896]  
Right: 282.1160278728621  
Intercept 281.65350838234457  
Prediction\_local [281.90161727]  
Right: 281.80361509317385  
Intercept 281.65695300890735  
Prediction\_local [281.64143412]  
Right: 281.5803869727943  
Intercept 281.59419424407645  
Prediction\_local [281.81477701]  
Right: 281.44759596820677  
Intercept 281.8869720018913  
Prediction\_local [281.06091725]  
Right: 281.7404934435435  
Intercept 281.53209347748054  
Prediction\_local [281.43561174]  
Right: 281.0042547977994  
Intercept 281.5249831280236  
Prediction\_local [281.58354196]  
Right: 281.8102439279549  
Intercept 281.7555084931875  
Prediction\_local [281.02055098]  
Right: 282.0801274335584  
Intercept 281.54345319721784  
Prediction\_local [281.92555129]  
Right: 281.59021814070246  
Intercept 281.6600220128799  
Prediction\_local [281.35664116]  
Right: 281.66017059912576  
Intercept 282.00185244053455  
Prediction\_local [280.99389687]  
Right: 282.0688057444434  
Intercept 281.6271152467131  
Prediction\_local [280.90342636]  
Right: 280.83288404537564  
Intercept 281.558538495976  
Prediction\_local [281.68857536]  
Right: 281.22931598301795  
Intercept 281.57430812707645  
Prediction\_local [281.26136309]  
Right: 281.8135367359707  
Intercept 281.56447583355686  
Prediction\_local [281.39792399]  
Right: 281.2296799733226  
Intercept 281.5967454475168  
Prediction\_local [281.85228979]  
Right: 281.6411555896316  
Intercept 281.6510360750897  
Prediction\_local [281.63676815]  
Right: 281.03037464056536  
Intercept 281.8385831736505  
Prediction\_local [281.46654493]  
Right: 281.0601398687579  
Intercept 281.5759581340964  
Prediction\_local [281.89386933]  
Right: 281.9298274473707  
Intercept 281.6498253377313  
Prediction\_local [281.99313212]  
Right: 281.959798038699  
Intercept 281.87435743658983  
Prediction\_local [280.89457372]  
Right: 280.79184991607804  
Intercept 281.5566296900755  
Prediction\_local [281.43714164]  
Right: 281.95007854965934  
Intercept 281.56202135477463  
Prediction\_local [281.32428299]  
Right: 281.31962403633645  
Intercept 281.93676453637835  
Prediction\_local [281.3487196]  
Right: 281.1032751431194  
Intercept 281.56594522243984  
Prediction\_local [281.58739939]  
Right: 281.21007347478206  
Intercept 281.6461783555253  
Prediction\_local [281.43349429]  
Right: 280.86004685386746  
Intercept 281.6713527761665  
Prediction\_local [281.33846643]  
Right: 281.1067104959993  
Intercept 281.75960642031043  
Prediction\_local [280.93129629]  
Right: 280.82865746217374  
Intercept 281.66142419100385  
Prediction\_local [281.02804001]  
Right: 280.9769045731093  
Intercept 281.68458473719465  
Prediction\_local [281.39903878]  
Right: 280.7701928540539  
Intercept 281.7301431314544  
Prediction\_local [281.83516512]  
Right: 281.5199632883228  
Intercept 281.6517240179861  
Prediction\_local [281.47951605]  
Right: 281.31957683236556  
Intercept 281.6619035100988  
Prediction\_local [281.40255044]  
Right: 281.4335965292003  
Intercept 281.47499510208837  
Prediction\_local [281.94951134]  
Right: 281.51142541850317  
Intercept 281.5407582563826  
Prediction\_local [281.58479052]  
Right: 280.88022530944056  
Intercept 281.605498346772  
Prediction\_local [281.26327316]  
Right: 281.2899031617253  
Intercept 281.57477287340237  
Prediction\_local [281.1410516]  
Right: 280.60982894644496  
Intercept 281.8769146950699  
Prediction\_local [281.34212335]  
Right: 280.97080713828336  
Intercept 281.92286027792403  
Prediction\_local [281.13003634]  
Right: 282.033329623539  
Intercept 281.6064879309395  
Prediction\_local [281.53290565]  
Right: 280.925462161  
Intercept 281.67529823343995  
Prediction\_local [280.97498995]  
Right: 280.72016547738883  
Intercept 281.4467080792501  
Prediction\_local [281.48800284]  
Right: 281.24015684144297  
Intercept 281.5556552109313  
Prediction\_local [281.99002674]  
Right: 281.7623367372036  
Intercept 281.3051518530602  
Prediction\_local [282.28028675]  
Right: 281.6037311309788  
Intercept 281.7223012761609  
Prediction\_local [281.12839713]  
Right: 282.03542566911284  
Intercept 281.7540502703328  
Prediction\_local [281.15672091]  
Right: 281.779564807586  
Intercept 281.6692767919933  
Prediction\_local [282.05943886]  
Right: 281.8895933588496  
Intercept 281.3890303246246  
Prediction\_local [282.05959911]  
Right: 281.2798741441664

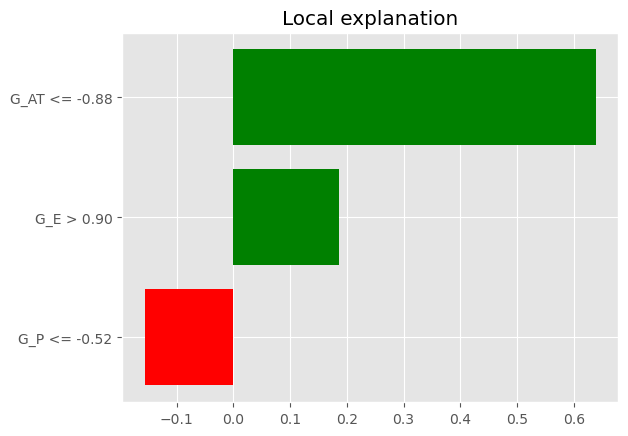
exp

<lime.explanation.Explanation at 0x25d904d0f40>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



# Plotting, as a bar chart, the actual global weights from the linear regression  
#with plt.style.context('ggplot'):  
# fig = plt.figure(figsize = (8,5))  
# plt.barh(range(len(model.coef\_)), model.coef\_, color = ['red' if coef < 0 else 'green' for coef in model.coef\_])  
# plt.yticks(range(len(model.coef\_)), X\_G\_train.columns);  
# plt.title('Weights')

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= -0.88', 0.6391379257068605),  
 ('G\_E > 0.90', 0.18699940397885728),  
 ('G\_P <= -0.52', -0.1555685462350835)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.6391379257068605),  
 (0, -0.18699940397885728),  
 (2, 0.1555685462350835)],  
 1: [(1, 0.6391379257068605),  
 (0, 0.18699940397885728),  
 (2, -0.1555685462350835)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [282.05959911]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.2798741441664

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr2GR\_train.html')

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, svr2.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 281.6248895450545  
Prediction\_local [282.02263902]  
Right: 281.69353806738343  
Intercept 281.73243863410454  
Prediction\_local [281.97240276]  
Right: 281.76251459515674  
Intercept 281.57626828511115  
Prediction\_local [281.35951084]  
Right: 281.0563244978809  
Intercept 281.8056423031455  
Prediction\_local [281.52582878]  
Right: 281.7193290811997  
Intercept 281.6900629495011  
Prediction\_local [281.20546921]  
Right: 281.28591565804174  
Intercept 281.62630785405753  
Prediction\_local [281.79021798]  
Right: 281.6004427145675  
Intercept 281.7299721145051  
Prediction\_local [280.889413]  
Right: 280.4066139846593  
Intercept 281.42627698255785  
Prediction\_local [282.03519694]  
Right: 281.7019292340987  
Intercept 281.2572071640744  
Prediction\_local [281.69025279]  
Right: 283.68472841279373  
Intercept 281.57662087172844  
Prediction\_local [281.62703829]  
Right: 281.648219329977  
Intercept 281.6624756248527  
Prediction\_local [281.42298893]  
Right: 281.0834846225682  
Intercept 281.9708416783374  
Prediction\_local [281.74172114]  
Right: 282.5599465610165  
Intercept 281.6449083332865  
Prediction\_local [281.81607633]  
Right: 281.64416159496386  
Intercept 281.1826989963295  
Prediction\_local [281.78694723]  
Right: 282.13901080276065  
Intercept 281.27217379615286  
Prediction\_local [281.32845172]  
Right: 281.23220243715184  
Intercept 281.7714492673965  
Prediction\_local [280.86542154]  
Right: 280.8645246017066  
Intercept 281.6172717047364  
Prediction\_local [280.94553776]  
Right: 281.6588831719554  
Intercept 281.79342916418926  
Prediction\_local [281.12827446]  
Right: 281.82472465794984  
Intercept 281.3334451692976  
Prediction\_local [282.23932707]  
Right: 283.47365893143336  
Intercept 281.6671481063681  
Prediction\_local [280.9645817]  
Right: 280.8521486353392  
Intercept 281.68576357570936  
Prediction\_local [281.52403668]  
Right: 282.0703346613765  
Intercept 281.60056714741455  
Prediction\_local [281.63675395]  
Right: 281.2206693531468  
Intercept 281.51414815101793  
Prediction\_local [281.84474293]  
Right: 281.98046444273183  
Intercept 281.58532143516067  
Prediction\_local [281.8788066]  
Right: 281.8917069558589  
Intercept 281.3766695899536  
Prediction\_local [281.88422994]  
Right: 281.97037203149085  
Intercept 281.51312783658597  
Prediction\_local [281.40153487]  
Right: 281.05976782162384  
Intercept 281.5184276068576  
Prediction\_local [282.69270859]  
Right: 285.72518863622304

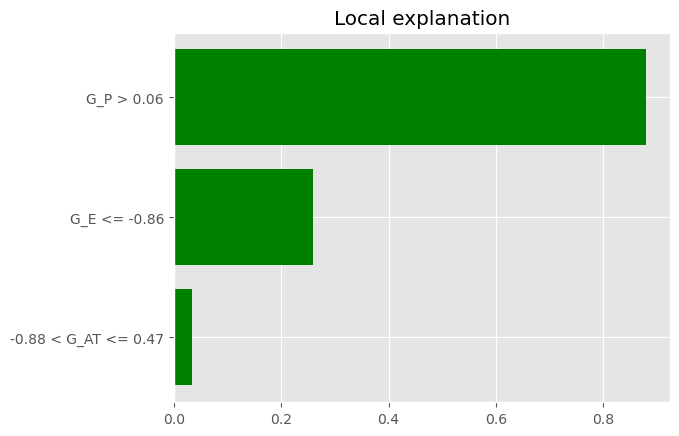
exp

<lime.explanation.Explanation at 0x25d9045f700>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_P > 0.06', 0.8799938574768997),  
 ('G\_E <= -0.86', 0.25969356221883994),  
 ('-0.88 < G\_AT <= 0.47', 0.03459356659663398)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.8799938574768997),  
 (0, -0.25969356221883994),  
 (1, -0.03459356659663398)],  
 1: [(2, 0.8799938574768997),  
 (0, 0.25969356221883994),  
 (1, 0.03459356659663398)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Loacl and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [282.69270859]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 285.72518863622304

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr2GR\_test.html')

###################################################################################################################

###################################################################################################################

###################################################################################################################

############ MODEL 3: RS Features (X\_R) and GT Target (LL\_G)###############

########### On the whole dataset  
model = SVR()  
svr\_param\_grid = {'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],  
 'epsilon' : [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'C' : [1, 30, 60, 90, 120, 150, 180]  
 }  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr3 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
svr3.fit(X\_R, y\_G)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [1, 30, 60, 90, 120, 150, 180],  
 'epsilon': [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'kernel': ['linear', 'rbf', 'poly', 'sigmoid']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The svr best parameters are:', svr3.best\_params\_)  
print('The svr best score is:', svr3.best\_score\_)

The svr best parameters are: {'C': 30, 'epsilon': 0.1, 'kernel': 'rbf'}  
The svr best score is: -0.15644768973917458

# The svr best parameters are: {'C': 30, 'epsilon': 0.1, 'kernel': 'rbf'}  
# The svr best score is: -0.15644768973917458

# Evaluation of the performance of the model on the whole dataset:  
svr\_y\_pred = svr3.predict(X\_R)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_G, svr\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_G, svr\_y\_pred))  
print('R2 Score:', r2\_score(y\_G, svr\_y\_pred))

Mean Absolute Error: 0.2551444362859172  
Mean Squared Error: 0.11572585547721483  
R2 Score: 0.6193324628876921

#Mean Absolute Error: 0.2551444362859172  
#Mean Squared Error: 0.11572585547721483  
#R2 Score: 0.6193324628876921

############## LIME ##############

###### Initializing the explainer using X\_R  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the full XG datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, svr3.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 279.41335829800477  
Prediction\_local [280.1039142]  
Right: 280.14440465676506  
Intercept 279.41768977907856  
Prediction\_local [279.82841194]  
Right: 280.03296453153666  
Intercept 279.4237550703757  
Prediction\_local [279.89913234]  
Right: 279.8598996006061  
Intercept 279.56330554062635  
Prediction\_local [279.82173321]  
Right: 279.7491074265906  
Intercept 279.5454336294063  
Prediction\_local [279.55408152]  
Right: 279.7427458366914  
Intercept 279.6012580290511  
Prediction\_local [279.44703864]  
Right: 279.29948925806485  
Intercept 279.86201216700493  
Prediction\_local [278.93596475]  
Right: 279.12375673403875  
Intercept 279.73180633744727  
Prediction\_local [278.93038822]  
Right: 279.04003939041763  
Intercept 279.7438470335819  
Prediction\_local [279.25430679]  
Right: 279.62998044101136  
Intercept 279.3259528993323  
Prediction\_local [280.08821202]  
Right: 280.3137345109421  
Intercept 279.2823557147185  
Prediction\_local [280.18660993]  
Right: 280.16218552352717  
Intercept 279.4787326539457  
Prediction\_local [280.15007826]  
Right: 280.23038906315895  
Intercept 279.4016783421544  
Prediction\_local [279.83117724]  
Right: 280.20124030202754  
Intercept 279.475814294298  
Prediction\_local [280.01106958]  
Right: 280.11817125091767  
Intercept 279.5670033569318  
Prediction\_local [279.68761751]  
Right: 279.8805965388495  
Intercept 279.5844025642684  
Prediction\_local [279.31167011]  
Right: 279.7578731371471  
Intercept 279.5051483102126  
Prediction\_local [279.51803065]  
Right: 279.21318750740124  
Intercept 279.4450223356451  
Prediction\_local [279.36921954]  
Right: 279.17001793338824  
Intercept 279.7008244210391  
Prediction\_local [279.2536869]  
Right: 278.9877520621709  
Intercept 279.72969097911033  
Prediction\_local [278.99982249]  
Right: 279.0632201598738  
Intercept 279.4044599547004  
Prediction\_local [279.80853973]  
Right: 279.55617581140706  
Intercept 279.46344342010207  
Prediction\_local [279.68383566]  
Right: 279.8204825314615  
Intercept 279.5101959420823  
Prediction\_local [279.83581074]  
Right: 280.225872493012  
Intercept 279.4331513444538  
Prediction\_local [280.00391247]  
Right: 280.234641744842  
Intercept 279.3488629798538  
Prediction\_local [280.17328603]  
Right: 280.35997962523  
Intercept 279.2465714439566  
Prediction\_local [280.02001035]  
Right: 280.03889349308906  
Intercept 279.5913830896068  
Prediction\_local [279.86417387]  
Right: 279.8948206900892  
Intercept 279.532902079608  
Prediction\_local [279.86827426]  
Right: 279.84651295809425  
Intercept 279.7380077784118  
Prediction\_local [279.31691619]  
Right: 279.6936037446277  
Intercept 279.62005531171775  
Prediction\_local [279.42185955]  
Right: 279.22464677052324  
Intercept 279.68863785576235  
Prediction\_local [279.06228874]  
Right: 279.0495283253364  
Intercept 279.68558239809556  
Prediction\_local [279.01537585]  
Right: 278.8828606548423  
Intercept 279.8209372415479  
Prediction\_local [279.17675413]  
Right: 279.45176625213173  
Intercept 279.56687839668297  
Prediction\_local [279.46365103]  
Right: 279.68004673228035  
Intercept 279.37309596766625  
Prediction\_local [280.09596739]  
Right: 280.2107107139176  
Intercept 279.3945918027127  
Prediction\_local [280.21305647]  
Right: 280.41035557913403  
Intercept 279.4417274926826  
Prediction\_local [280.17215052]  
Right: 280.4758280184377  
Intercept 279.3679497103636  
Prediction\_local [280.15264731]  
Right: 280.29467957747795  
Intercept 279.45886425182044  
Prediction\_local [279.75975763]  
Right: 279.8041794202058  
Intercept 279.63199090249844  
Prediction\_local [279.61366946]  
Right: 279.74513154655955  
Intercept 279.624245437471  
Prediction\_local [279.37415962]  
Right: 279.43721798372445  
Intercept 279.56394439263266  
Prediction\_local [279.4057898]  
Right: 279.1802862749662  
Intercept 279.7183260638464  
Prediction\_local [278.91862404]  
Right: 279.0205655885807  
Intercept 279.64565664476567  
Prediction\_local [278.99858679]  
Right: 278.9766896010098  
Intercept 279.78220266829334  
Prediction\_local [278.9035071]  
Right: 279.59957414984405  
Intercept 279.53716304801117  
Prediction\_local [279.49187239]  
Right: 279.9464319673779  
Intercept 279.3586509406798  
Prediction\_local [280.11861987]  
Right: 280.12445794293393  
Intercept 279.3588258664063  
Prediction\_local [280.07659194]  
Right: 280.31817317394615  
Intercept 279.5013933899531  
Prediction\_local [280.01013101]  
Right: 280.3077150244769  
Intercept 279.39246138261325  
Prediction\_local [280.17240158]  
Right: 280.35516983530397  
Intercept 279.47499710458095  
Prediction\_local [279.91836185]  
Right: 280.0424082080224  
Intercept 279.7233739127072  
Prediction\_local [279.04343734]  
Right: 279.63485045326166  
Intercept 279.61743887298456  
Prediction\_local [279.54502313]  
Right: 279.56270122984427  
Intercept 279.7796189743772  
Prediction\_local [278.89943289]  
Right: 279.35510386070035  
Intercept 279.93816328725956  
Prediction\_local [278.69760069]  
Right: 278.9754477335655  
Intercept 279.7860849920352  
Prediction\_local [279.0138925]  
Right: 278.88235093829667  
Intercept 279.61985313474594  
Prediction\_local [279.37349943]  
Right: 279.28220413657465  
Intercept 279.5421833872298  
Prediction\_local [279.55837042]  
Right: 279.9586662966634  
Intercept 279.33404319208444  
Prediction\_local [279.95514497]  
Right: 280.1134110715578  
Intercept 279.4469581827081  
Prediction\_local [280.11950111]  
Right: 280.21652549308607  
Intercept 279.4248668632305  
Prediction\_local [280.18619706]  
Right: 280.3903912672444  
Intercept 279.4547829033086  
Prediction\_local [279.85877705]  
Right: 280.04034127557503  
Intercept 279.68640476972917  
Prediction\_local [279.2044627]  
Right: 279.51408910728196  
Intercept 279.5912899915257  
Prediction\_local [279.62125517]  
Right: 279.7181434094952  
Intercept 279.60940301804146  
Prediction\_local [279.36551033]  
Right: 279.30845604605935  
Intercept 279.67848974371793  
Prediction\_local [279.10437563]  
Right: 279.11021737250974  
Intercept 279.79557375708066  
Prediction\_local [278.9474284]  
Right: 279.02997030861934  
Intercept 279.79320328951866  
Prediction\_local [278.98470887]  
Right: 279.5364408536848  
Intercept 279.73721461570386  
Prediction\_local [278.90724292]  
Right: 279.46069333870395  
Intercept 279.5556619661885  
Prediction\_local [279.64655781]  
Right: 280.0542595839331  
Intercept 279.4171021802523  
Prediction\_local [280.05134232]  
Right: 280.1034521780307  
Intercept 279.37573217552193  
Prediction\_local [280.21684571]  
Right: 280.3519817724888  
Intercept 279.3371151161907  
Prediction\_local [280.29122965]  
Right: 280.3207486425414  
Intercept 279.3092721632353  
Prediction\_local [280.14465015]  
Right: 280.28400915776103  
Intercept 279.52047275116206  
Prediction\_local [279.7263571]  
Right: 279.9249274377234  
Intercept 279.6265810032891  
Prediction\_local [279.37879102]  
Right: 279.601844071393  
Intercept 279.53333403264014  
Prediction\_local [279.36879098]  
Right: 278.76860587699133  
Intercept 279.81253413032164  
Prediction\_local [278.93528725]  
Right: 279.29030701831863  
Intercept 279.78678950896716  
Prediction\_local [278.84798297]  
Right: 278.9520031090918  
Intercept 279.7826444188774  
Prediction\_local [279.05537898]  
Right: 279.047132957148  
Intercept 279.84476691462277  
Prediction\_local [278.8418005]  
Right: 279.33211846856994  
Intercept 279.72477307908514  
Prediction\_local [278.82643511]  
Right: 279.62718105735837  
Intercept 279.4543198577995  
Prediction\_local [280.176266]  
Right: 280.23425261243733  
Intercept 279.29926108591167  
Prediction\_local [280.17296683]  
Right: 280.3986188544333  
Intercept 279.31158410528326  
Prediction\_local [279.98283112]  
Right: 280.4387665134026  
Intercept 279.4639714830563  
Prediction\_local [280.06059885]  
Right: 280.3345889669297  
Intercept 279.39390027924304  
Prediction\_local [279.94899423]  
Right: 279.97599115880195  
Intercept 279.42059099527796  
Prediction\_local [279.74356915]  
Right: 279.7489079294281  
Intercept 279.53294664261733  
Prediction\_local [279.60914237]  
Right: 279.75745938957454  
Intercept 279.5646935831259  
Prediction\_local [279.44618774]  
Right: 279.44237774553136  
Intercept 279.85675355972137  
Prediction\_local [278.88493974]  
Right: 279.55024010129654  
Intercept 279.7783922706882  
Prediction\_local [279.12533812]  
Right: 279.56046308043364  
Intercept 279.72510743116  
Prediction\_local [278.9248115]  
Right: 279.22579093242484  
Intercept 279.6007728496751  
Prediction\_local [279.57040911]  
Right: 280.0651513586571  
Intercept 279.4626724198173  
Prediction\_local [279.98264425]  
Right: 280.2125406669447  
Intercept 279.37722389379337  
Prediction\_local [280.20882935]  
Right: 280.2552019242695  
Intercept 279.3996905566134  
Prediction\_local [280.18384155]  
Right: 280.3023366003919  
Intercept 279.34840351881536  
Prediction\_local [280.02524592]  
Right: 280.32122621806036  
Intercept 279.36017919278436  
Prediction\_local [280.10494355]  
Right: 280.3468943556983  
Intercept 279.4654411351708  
Prediction\_local [279.80673898]  
Right: 280.01033225635933  
Intercept 279.4764761919074  
Prediction\_local [279.87659588]  
Right: 279.89978435722804  
Intercept 279.3888169806115  
Prediction\_local [279.88130209]  
Right: 280.07798040478946  
Intercept 279.7390756319646  
Prediction\_local [279.02982123]  
Right: 278.8903261005039  
Intercept 279.61644833018016  
Prediction\_local [279.28544555]  
Right: 279.50175615027985  
Intercept 279.6827682756364  
Prediction\_local [279.30434392]  
Right: 279.4299869320726  
Intercept 279.35233867916037  
Prediction\_local [280.14500154]  
Right: 280.2402946073816  
Intercept 279.4343707514729  
Prediction\_local [279.89662876]  
Right: 280.12868431798876  
Intercept 279.42196077473574  
Prediction\_local [280.1438346]  
Right: 280.4035016321499

exp

<lime.explanation.Explanation at 0x24a276d0810>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('0.05 < P\_R <= 3.71', 0.4796862201674235),  
 ('AT\_R <= 26.86', 0.2938570416089151),  
 ('0.79 < E\_R <= 7.12', -0.051669437050488695)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.4796862201674235),  
 (1, -0.2938570416089151),  
 (0, 0.051669437050488695)],  
 1: [(2, 0.4796862201674235),  
 (1, 0.2938570416089151),  
 (0, -0.051669437050488695)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.1438346]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.4035016321499

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr3RG\_Whole.html')

#################################################################################################

########### Fitting the model on the training dataset using the best parameters  
model = SVR()  
svr\_param\_grid = {'kernel': ['rbf'],  
 'epsilon' : [0.1],  
 'C' : [30]  
 }  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr3 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1 )  
svr3.fit(X\_R\_train, y\_G\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [30], 'epsilon': [0.1], 'kernel': ['rbf']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The svr best score is:', svr3.best\_score\_)

The svr best score is: -0.19305540264730786

# Evaluation of the performance of the model on the training dataset:  
svr\_y\_predtr = svr3.predict(X\_R\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, svr\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, svr\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, svr\_y\_predtr))

The training MAE is: 0.19969555244281914  
The training MSE is: 0.07882082025146947  
The training R2 Score is: 0.7414830530544194

# Evaluation of the performance of the model on the testing dataset:  
svr\_y\_predts = svr3.predict(X\_R\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, svr\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, svr\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, svr\_y\_predts))

The testing MAE is: 0.3565370989352893  
The testing MSE is: 0.25012490593793624  
The testing R2 Score is: 0.08790799084434175

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_svr3tr = cross\_val\_score(svr3, X\_R\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_svr3tr)))

The mean training CV score is: 0.16692160277435897

############# On the testing dataset   
score\_svr3ts = cross\_val\_score(svr3, X\_R\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_svr3ts)))

The mean testing CV score is: 0.1920870783196603

############## LIME ##############

###### Initializing the explainer using the training dataset  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, svr3.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 279.2166396820286  
Prediction\_local [279.63503254]  
Right: 279.879579664249  
Intercept 279.107774587643  
Prediction\_local [279.894811]  
Right: 279.75989379355167  
Intercept 279.35810312658253  
Prediction\_local [279.48163578]  
Right: 280.30221500014665  
Intercept 279.57013041233745  
Prediction\_local [278.9710932]  
Right: 278.7400494265628  
Intercept 279.67619557621265  
Prediction\_local [278.1917407]  
Right: 278.83974862927073  
Intercept 279.3077174949565  
Prediction\_local [279.92865484]  
Right: 280.32119136506446  
Intercept 279.7270117954567  
Prediction\_local [278.95708224]  
Right: 279.35013955557486  
Intercept 279.36986206736225  
Prediction\_local [279.57720777]  
Right: 279.68290929987916  
Intercept 279.28238742572256  
Prediction\_local [279.48869211]  
Right: 279.49149038006504  
Intercept 279.63781852762816  
Prediction\_local [278.80103454]  
Right: 279.1572067679563  
Intercept 279.5433459781789  
Prediction\_local [278.90082789]  
Right: 279.8240016964593  
Intercept 279.2277127334821  
Prediction\_local [279.80001048]  
Right: 279.4909159962254  
Intercept 279.32748770119275  
Prediction\_local [279.54027543]  
Right: 280.0142292032253  
Intercept 279.66915071770035  
Prediction\_local [278.60398263]  
Right: 279.24957251115137  
Intercept 279.2946376482421  
Prediction\_local [279.63759802]  
Right: 280.1169241530983  
Intercept 279.667204339833  
Prediction\_local [278.64228399]  
Right: 279.0301532134544  
Intercept 279.17862627737355  
Prediction\_local [279.77187106]  
Right: 280.36874724508385  
Intercept 279.58670261101236  
Prediction\_local [278.86535871]  
Right: 279.8289444742803  
Intercept 279.5921270089145  
Prediction\_local [279.46000958]  
Right: 279.18982209384126  
Intercept 279.6269768200434  
Prediction\_local [278.61063409]  
Right: 279.69379930518505  
Intercept 279.71503871769994  
Prediction\_local [278.33236864]  
Right: 279.04401208744935  
Intercept 279.25883783078007  
Prediction\_local [280.1017789]  
Right: 280.10765883118574  
Intercept 279.33427618220406  
Prediction\_local [279.94307431]  
Right: 279.7823084874297  
Intercept 279.54243084032225  
Prediction\_local [279.2962716]  
Right: 279.4804095323609  
Intercept 279.3720557479472  
Prediction\_local [279.75991064]  
Right: 279.7886442093357  
Intercept 279.3031350941522  
Prediction\_local [280.00699497]  
Right: 280.160260567927  
Intercept 279.64627231910345  
Prediction\_local [279.1956301]  
Right: 279.2995353218817  
Intercept 279.14261735617583  
Prediction\_local [279.73957974]  
Right: 280.149473185573  
Intercept 279.59437172961907  
Prediction\_local [278.53703922]  
Right: 278.76018548181867  
Intercept 279.255696569469  
Prediction\_local [279.87488024]  
Right: 280.1400804205299  
Intercept 279.3370159577511  
Prediction\_local [279.78083858]  
Right: 279.56699872054367  
Intercept 279.21927631559254  
Prediction\_local [279.94583514]  
Right: 280.38019976450266  
Intercept 279.41231796100016  
Prediction\_local [279.65170278]  
Right: 279.89006744947727  
Intercept 279.146161937375  
Prediction\_local [280.0572644]  
Right: 280.370178219001  
Intercept 279.34296860440736  
Prediction\_local [279.7651696]  
Right: 280.1795571951758  
Intercept 279.5082089607638  
Prediction\_local [278.76902786]  
Right: 279.5213846090588  
Intercept 279.24514782158695  
Prediction\_local [279.89846838]  
Right: 280.18008313756627  
Intercept 279.12342324886924  
Prediction\_local [280.11872588]  
Right: 280.1498473675873  
Intercept 279.1136090189313  
Prediction\_local [280.00659008]  
Right: 279.5025508650573  
Intercept 279.2403587714715  
Prediction\_local [279.43923882]  
Right: 280.2098768228926  
Intercept 279.096678757494  
Prediction\_local [280.18938375]  
Right: 280.1714653797808  
Intercept 279.4603985051291  
Prediction\_local [279.26299609]  
Right: 279.59982381228696  
Intercept 279.2626723397003  
Prediction\_local [280.306751]  
Right: 279.85453596931296  
Intercept 279.0778565396288  
Prediction\_local [280.05631335]  
Right: 280.22681970968307  
Intercept 279.6923547091576  
Prediction\_local [279.27043235]  
Right: 279.54210091037305  
Intercept 279.71303254380194  
Prediction\_local [278.95020015]  
Right: 279.0794846122096  
Intercept 279.369889983342  
Prediction\_local [279.51530406]  
Right: 279.74985973254456  
Intercept 279.3267901955562  
Prediction\_local [279.82518149]  
Right: 280.1883936573786  
Intercept 279.38435545049987  
Prediction\_local [279.78872938]  
Right: 280.2775828119379  
Intercept 279.5667620041935  
Prediction\_local [278.67058605]  
Right: 279.09040946157455  
Intercept 279.3355134922111  
Prediction\_local [279.70778187]  
Right: 279.5826957517391  
Intercept 279.28387432224315  
Prediction\_local [279.92633994]  
Right: 280.2998250825587  
Intercept 279.1670642695924  
Prediction\_local [279.91805837]  
Right: 279.9423423784339  
Intercept 279.4400651105943  
Prediction\_local [279.32972734]  
Right: 279.4603924945579  
Intercept 279.4917216720766  
Prediction\_local [279.00671912]  
Right: 279.72599047030985  
Intercept 279.25507571070375  
Prediction\_local [279.92018927]  
Right: 280.3900703407253  
Intercept 279.26346540790786  
Prediction\_local [279.54513788]  
Right: 279.80631860245745  
Intercept 279.3847692066524  
Prediction\_local [279.30731685]  
Right: 279.54082614653987  
Intercept 279.5895499261175  
Prediction\_local [278.67237071]  
Right: 279.17866922024336  
Intercept 279.46135288016757  
Prediction\_local [279.43746769]  
Right: 279.20034438126544  
Intercept 279.4036594296686  
Prediction\_local [279.17238832]  
Right: 279.2602044442611  
Intercept 279.42907602714314  
Prediction\_local [279.72026197]  
Right: 279.7169238156463  
Intercept 279.37469847495277  
Prediction\_local [279.14466836]  
Right: 278.84987934194453  
Intercept 279.24379057193073  
Prediction\_local [280.02347941]  
Right: 280.02043770191614  
Intercept 279.3350208145831  
Prediction\_local [279.71988927]  
Right: 280.14510693260115  
Intercept 279.65231899500344  
Prediction\_local [279.10841505]  
Right: 279.65106823085637  
Intercept 279.2843462448291  
Prediction\_local [279.65721672]  
Right: 280.0740443590075  
Intercept 279.5139630025781  
Prediction\_local [279.3243463]  
Right: 279.11042423595  
Intercept 279.6897107711615  
Prediction\_local [278.68733483]  
Right: 279.0397694726523  
Intercept 279.45859060509537  
Prediction\_local [279.31144329]  
Right: 279.0265797176023  
Intercept 279.18297672641273  
Prediction\_local [279.68293568]  
Right: 279.55235905640154  
Intercept 279.2595517764898  
Prediction\_local [279.98528699]  
Right: 280.2514646872582  
Intercept 279.3618402993938  
Prediction\_local [279.26334982]  
Right: 279.1610494049792  
Intercept 279.5864625824694  
Prediction\_local [278.70119062]  
Right: 279.01193766134946  
Intercept 279.377227861028  
Prediction\_local [279.4868143]  
Right: 279.6830307049968  
Intercept 279.117307583095  
Prediction\_local [279.86511734]  
Right: 280.3034983233959  
Intercept 279.34155419121646  
Prediction\_local [279.8527846]  
Right: 280.2914072993217  
Intercept 279.40366954354835  
Prediction\_local [279.62985267]  
Right: 279.9570703965656  
Intercept 279.3225999726061  
Prediction\_local [279.44886802]  
Right: 280.00973973430655  
Intercept 279.37129209138647  
Prediction\_local [279.62409306]  
Right: 280.1801246882543  
Intercept 279.2706048535997  
Prediction\_local [279.80982153]  
Right: 280.18233567892815

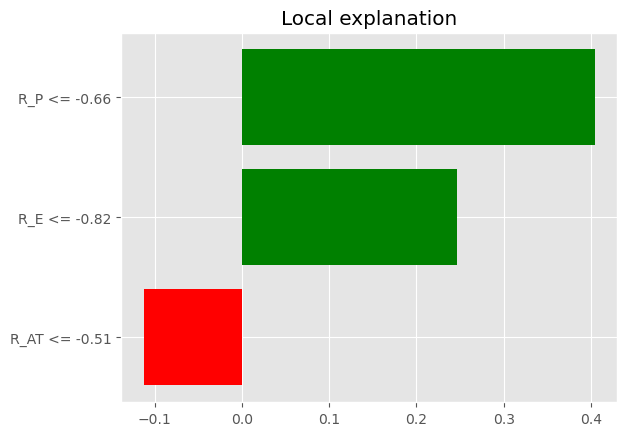
exp

<lime.explanation.Explanation at 0x25d90533820>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P <= -0.66', 0.4044713542432914),  
 ('R\_E <= -0.82', 0.2468825339988655),  
 ('R\_AT <= -0.51', -0.11213720904715622)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.4044713542432914),  
 (0, -0.2468825339988655),  
 (1, 0.11213720904715622)],  
 1: [(2, 0.4044713542432914),  
 (0, 0.2468825339988655),  
 (1, -0.11213720904715622)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.80982153]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.18233567892815

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr3RG\_train.html')

######################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, svr3.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.6458759393509  
Prediction\_local [279.736672]  
Right: 280.1208331495276  
Intercept 279.3441759934147  
Prediction\_local [279.86976588]  
Right: 280.3032027578335  
Intercept 279.4562403370437  
Prediction\_local [279.55697774]  
Right: 279.50634130589754  
Intercept 279.50191569256657  
Prediction\_local [278.79486402]  
Right: 278.3796270445602  
Intercept 279.2393680567942  
Prediction\_local [279.60255204]  
Right: 279.21449043543214  
Intercept 279.3421856727774  
Prediction\_local [279.20514196]  
Right: 278.136118030661  
Intercept 279.52352809773  
Prediction\_local [279.30062011]  
Right: 279.3848535523384  
Intercept 279.36021011179855  
Prediction\_local [279.34563062]  
Right: 280.1181210263202  
Intercept 279.312526826149  
Prediction\_local [278.88151271]  
Right: 279.0705564073861  
Intercept 279.20922733376375  
Prediction\_local [280.06806253]  
Right: 280.05653185099203  
Intercept 279.3603705464069  
Prediction\_local [279.48071025]  
Right: 279.5281945612012  
Intercept 279.2413724734472  
Prediction\_local [280.19168472]  
Right: 280.75739693471456  
Intercept 279.357906144313  
Prediction\_local [280.21489365]  
Right: 280.1125736443275  
Intercept 279.19004721425705  
Prediction\_local [279.93570363]  
Right: 280.2310746113288  
Intercept 279.31247682659057  
Prediction\_local [279.58205407]  
Right: 279.57146008801567  
Intercept 279.51949828401416  
Prediction\_local [279.47140564]  
Right: 279.5645829040413  
Intercept 279.46921892389764  
Prediction\_local [279.42773145]  
Right: 279.64161825462344  
Intercept 279.4250289090036  
Prediction\_local [279.1160835]  
Right: 279.12986325531176  
Intercept 279.4502671280813  
Prediction\_local [278.67496164]  
Right: 279.2861109420655  
Intercept 279.51375648739184  
Prediction\_local [279.33992804]  
Right: 279.48274649652916  
Intercept 279.4776305060214  
Prediction\_local [279.81864924]  
Right: 280.1977384609645  
Intercept 279.51058467876476  
Prediction\_local [278.72623813]  
Right: 278.540071434129  
Intercept 279.3613965004017  
Prediction\_local [279.55194849]  
Right: 279.9925632169919  
Intercept 279.60762422999414  
Prediction\_local [279.40658654]  
Right: 280.097858836778  
Intercept 279.16437398754243  
Prediction\_local [279.92931472]  
Right: 280.2732132756588  
Intercept 279.41084424695435  
Prediction\_local [279.25708992]  
Right: 279.443611269453  
Intercept 279.8043454194289  
Prediction\_local [278.63497389]  
Right: 279.46965941743827

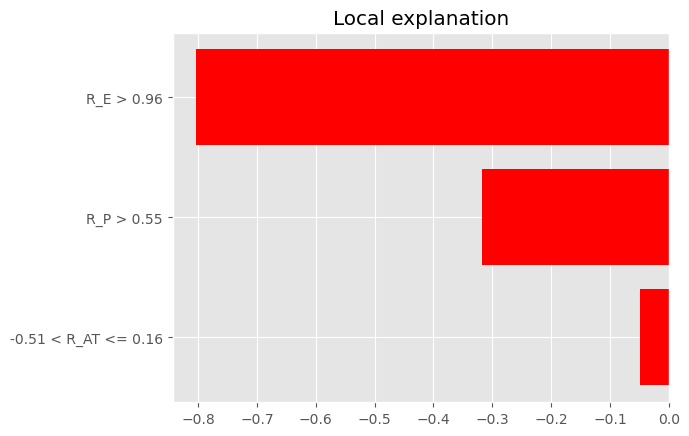
exp

<lime.explanation.Explanation at 0x25d90ba37f0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_E > 0.96', -0.8026931574001123),  
 ('R\_P > 0.55', -0.3176345351158108),  
 ('-0.51 < R\_AT <= 0.16', -0.049043833743056105)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(0, 0.8026931574001123),  
 (2, 0.3176345351158108),  
 (1, 0.049043833743056105)],  
 1: [(0, -0.8026931574001123),  
 (2, -0.3176345351158108),  
 (1, -0.049043833743056105)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [278.63497389]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.46965941743827

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr3RG\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

############ MODEL 4: GT Features (X\_G) and GT Target (LL\_G)###############

########### On the whole dataset  
model = SVR()  
svr\_param\_grid = {'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],  
 'epsilon' : [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'C' : [1, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330]  
 }  
  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr4 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
svr4.fit(X\_G, y\_G)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [1, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300,  
 330],  
 'epsilon': [0.0001, 0.001, 0.01, 0.1, 1, 10],  
 'kernel': ['linear', 'rbf', 'poly', 'sigmoid']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The svr best parameters are:', svr4.best\_params\_)  
print('The svr best score is:', svr4.best\_score\_)

The svr best parameters are: {'C': 300, 'epsilon': 0.1, 'kernel': 'rbf'}  
The svr best score is: -0.12362689221798343

#The svr best parameters are: {'C': 300, 'epsilon': 0.1, 'kernel': 'rbf'}  
#The svr best score is: -0.12362689221798343

# Evaluation of the performance of the model on the whole dataset:  
svr\_y\_pred = svr4.predict(X\_G)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_G, svr\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_G, svr\_y\_pred))  
print('R2 Score:', r2\_score(y\_G, svr\_y\_pred))

Mean Absolute Error: 0.20728904669257198  
Mean Squared Error: 0.07848598328169103  
R2 Score: 0.7418289471226975

############### LIME #################

###### Initializing the explainer using the whole X\_G  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the whole XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, svr4.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 279.4276123494396  
Prediction\_local [280.57285284]  
Right: 280.35051714966573  
Intercept 278.9082895670983  
Prediction\_local [280.2792611]  
Right: 280.28988519138755  
Intercept 279.3921973666535  
Prediction\_local [279.65602922]  
Right: 279.7019242458174  
Intercept 279.4714735728403  
Prediction\_local [279.50719831]  
Right: 279.4947810701676  
Intercept 279.411540396917  
Prediction\_local [279.61299867]  
Right: 279.57752210432653  
Intercept 280.0347674438987  
Prediction\_local [279.36962574]  
Right: 279.5118662149207  
Intercept 279.82493398220777  
Prediction\_local [279.29939912]  
Right: 279.34876450756644  
Intercept 279.8635850481473  
Prediction\_local [279.21494618]  
Right: 279.21455065487913  
Intercept 279.882213103378  
Prediction\_local [279.37477091]  
Right: 279.68695712003495  
Intercept 279.2163778310167  
Prediction\_local [280.12172778]  
Right: 280.01152852463014  
Intercept 279.183106940604  
Prediction\_local [280.36343407]  
Right: 280.47438688023374  
Intercept 279.07342473608617  
Prediction\_local [280.18120988]  
Right: 280.4478564870914  
Intercept 278.9704962944734  
Prediction\_local [280.43147669]  
Right: 280.29195948782234  
Intercept 279.3411633695295  
Prediction\_local [280.23443105]  
Right: 280.2498564416256  
Intercept 279.49559550671694  
Prediction\_local [279.7087913]  
Right: 279.8267893198639  
Intercept 279.32528446001703  
Prediction\_local [279.49704387]  
Right: 279.56329085382725  
Intercept 279.43438980000417  
Prediction\_local [279.54292434]  
Right: 279.6119712580146  
Intercept 279.4971950669183  
Prediction\_local [279.50296838]  
Right: 279.57803403308134  
Intercept 280.074985284012  
Prediction\_local [278.89590984]  
Right: 279.04521894904644  
Intercept 280.1177433690831  
Prediction\_local [278.66690568]  
Right: 279.0299848034969  
Intercept 279.8081707123947  
Prediction\_local [279.53764281]  
Right: 279.49977069838235  
Intercept 279.2570647488382  
Prediction\_local [280.32541909]  
Right: 280.0669090129848  
Intercept 279.1681503607811  
Prediction\_local [280.06960643]  
Right: 280.3879354901748  
Intercept 279.17259294227205  
Prediction\_local [280.50107739]  
Right: 280.3902013421945  
Intercept 279.0205922422365  
Prediction\_local [280.34617865]  
Right: 280.25977869437423  
Intercept 279.4500807538644  
Prediction\_local [280.35871557]  
Right: 280.19895682835914  
Intercept 279.31718325281594  
Prediction\_local [279.55250171]  
Right: 279.72274631666494  
Intercept 279.38031676187677  
Prediction\_local [279.52820434]  
Right: 279.6061820391185  
Intercept 279.28018165347316  
Prediction\_local [279.80415642]  
Right: 279.54897437234166  
Intercept 280.2111983918827  
Prediction\_local [278.48721149]  
Right: 279.04445027396173  
Intercept 279.85013332132354  
Prediction\_local [279.55884484]  
Right: 279.25223566266015  
Intercept 279.6882341402089  
Prediction\_local [279.39013144]  
Right: 279.11686152877684  
Intercept 279.8237508936539  
Prediction\_local [279.43398065]  
Right: 279.12967952562593  
Intercept 279.0215675642454  
Prediction\_local [280.13729974]  
Right: 280.0012728874113  
Intercept 279.43539762685816  
Prediction\_local [280.07567778]  
Right: 280.12373730410485  
Intercept 279.0512050140872  
Prediction\_local [280.28946931]  
Right: 280.2135928686165  
Intercept 279.1834770398117  
Prediction\_local [280.30401989]  
Right: 280.13925656173564  
Intercept 278.9643913352216  
Prediction\_local [280.37897853]  
Right: 280.0784052011469  
Intercept 279.30933135000737  
Prediction\_local [279.55566942]  
Right: 279.79882609046564  
Intercept 279.7169430495891  
Prediction\_local [279.49190045]  
Right: 279.44454043728376  
Intercept 280.12898266750835  
Prediction\_local [278.90388133]  
Right: 279.2084455500236  
Intercept 280.1517939789134  
Prediction\_local [279.10117039]  
Right: 279.1445322425843  
Intercept 279.8965329699664  
Prediction\_local [279.38737948]  
Right: 279.0529670696896  
Intercept 279.874197544067  
Prediction\_local [279.40147275]  
Right: 278.9203514442984  
Intercept 279.66221426665913  
Prediction\_local [279.98528713]  
Right: 279.60041014175994  
Intercept 279.21085477360276  
Prediction\_local [279.95759044]  
Right: 280.0277036658747  
Intercept 278.87099035102517  
Prediction\_local [280.44274911]  
Right: 280.2799854087659  
Intercept 279.053852085956  
Prediction\_local [280.39172149]  
Right: 280.3237566748187  
Intercept 279.17688471439664  
Prediction\_local [280.35800234]  
Right: 280.2034778799921  
Intercept 279.15871409960783  
Prediction\_local [280.31403955]  
Right: 280.08071597624814  
Intercept 279.35708430085515  
Prediction\_local [279.91458834]  
Right: 279.9462022264942  
Intercept 279.44155962918353  
Prediction\_local [279.4334837]  
Right: 279.5718867422998  
Intercept 279.2796754356036  
Prediction\_local [279.80676799]  
Right: 279.6158237481772  
Intercept 279.7473034900037  
Prediction\_local [279.38864727]  
Right: 279.14281286579495  
Intercept 279.9408428821551  
Prediction\_local [279.15252551]  
Right: 278.9598446622841  
Intercept 279.96664315237336  
Prediction\_local [279.12599425]  
Right: 279.2307889691531  
Intercept 279.77219740409737  
Prediction\_local [279.4531477]  
Right: 279.4703564308713  
Intercept 279.4320054808843  
Prediction\_local [279.9898958]  
Right: 279.9800490602558  
Intercept 279.18048864082573  
Prediction\_local [280.41367956]  
Right: 280.42925756505133  
Intercept 279.0304817491424  
Prediction\_local [280.35156457]  
Right: 280.4231321365325  
Intercept 279.2128143353784  
Prediction\_local [280.49968234]  
Right: 280.0204044740002  
Intercept 279.0745506623957  
Prediction\_local [280.11696881]  
Right: 280.01447423794673  
Intercept 279.3269379301439  
Prediction\_local [279.63969418]  
Right: 279.7848415127116  
Intercept 279.3709263375632  
Prediction\_local [279.47510909]  
Right: 279.4231525271341  
Intercept 279.36764220923516  
Prediction\_local [279.6887257]  
Right: 279.55570511229524  
Intercept 279.99158590297685  
Prediction\_local [278.95452422]  
Right: 279.03602902172696  
Intercept 279.9908698491652  
Prediction\_local [279.28124984]  
Right: 279.0297114511511  
Intercept 279.9310500340759  
Prediction\_local [279.40836307]  
Right: 279.68978990654216  
Intercept 279.7920364203478  
Prediction\_local [279.4913341]  
Right: 279.72721124282464  
Intercept 279.82309428385355  
Prediction\_local [279.57773124]  
Right: 280.29637750017315  
Intercept 279.09862979411895  
Prediction\_local [280.21886512]  
Right: 280.2787046250959  
Intercept 279.0769337409856  
Prediction\_local [280.47140799]  
Right: 280.34523620918486  
Intercept 279.201256776453  
Prediction\_local [280.40001441]  
Right: 280.0769946392773  
Intercept 279.19149778671954  
Prediction\_local [280.24245666]  
Right: 280.06006275414853  
Intercept 279.33813922886804  
Prediction\_local [279.72941433]  
Right: 279.7890569437121  
Intercept 279.63109722446785  
Prediction\_local [279.37679725]  
Right: 279.5612063266296  
Intercept 279.4810964095244  
Prediction\_local [279.61007537]  
Right: 279.45968513775716  
Intercept 280.09626674963306  
Prediction\_local [278.79129645]  
Right: 278.81393102217436  
Intercept 279.8992932863239  
Prediction\_local [279.50389111]  
Right: 279.4003412714223  
Intercept 280.1086828143521  
Prediction\_local [279.22642701]  
Right: 279.3304895418277  
Intercept 279.85220392806093  
Prediction\_local [279.58364983]  
Right: 279.5869308652061  
Intercept 279.9274366215979  
Prediction\_local [279.4560008]  
Right: 280.2195688255846  
Intercept 279.12508575112975  
Prediction\_local [280.14114394]  
Right: 280.3463376986004  
Intercept 279.00041083293166  
Prediction\_local [280.55384217]  
Right: 280.3121951010745  
Intercept 279.15398664158954  
Prediction\_local [280.27205482]  
Right: 280.04209073379343  
Intercept 279.07864794116654  
Prediction\_local [280.33197445]  
Right: 280.05755867670047  
Intercept 279.31526439005864  
Prediction\_local [279.78300633]  
Right: 279.83717461841394  
Intercept 279.25942381459714  
Prediction\_local [279.40838164]  
Right: 279.49794218617564  
Intercept 279.1970964256228  
Prediction\_local [279.67600473]  
Right: 279.5956056431087  
Intercept 279.95049579159655  
Prediction\_local [279.18280867]  
Right: 279.38518889989155  
Intercept 279.93064518568104  
Prediction\_local [279.16528791]  
Right: 279.09498974648034  
Intercept 280.0643608102566  
Prediction\_local [278.9703562]  
Right: 279.3334760771793  
Intercept 279.8514462969874  
Prediction\_local [279.62564147]  
Right: 279.9996780899301  
Intercept 279.0029468515599  
Prediction\_local [280.43098605]  
Right: 280.29753317866425  
Intercept 279.22241621630064  
Prediction\_local [280.33157425]  
Right: 280.37848241108065  
Intercept 279.06216943771403  
Prediction\_local [280.47810186]  
Right: 280.451323540005  
Intercept 279.057200128755  
Prediction\_local [280.14490272]  
Right: 280.1068633615433  
Intercept 279.0125645711044  
Prediction\_local [280.28543344]  
Right: 280.14567057097577  
Intercept 279.3236464099365  
Prediction\_local [279.82006653]  
Right: 279.7613818225246  
Intercept 279.28981524391446  
Prediction\_local [279.52606201]  
Right: 279.5599359539242  
Intercept 279.54062637707636  
Prediction\_local [279.57353837]  
Right: 279.6394018326909  
Intercept 279.8318120271072  
Prediction\_local [279.32414764]  
Right: 279.70033032129845  
Intercept 280.099498449841  
Prediction\_local [279.06446995]  
Right: 279.09041064326533  
Intercept 279.9428594522715  
Prediction\_local [279.37542167]  
Right: 279.1251293005276  
Intercept 279.8121529377778  
Prediction\_local [279.57311682]  
Right: 279.10741470506804  
Intercept 279.0014054660809  
Prediction\_local [280.22155829]  
Right: 280.123562534461  
Intercept 279.18818934544197  
Prediction\_local [280.19809037]  
Right: 280.3665323385066  
Intercept 279.009300616343  
Prediction\_local [280.23714795]  
Right: 280.4046384308538

exp

<lime.explanation.Explanation at 0x24a27641b50>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('P\_G <= 0.00', 0.9694712330882354),  
 ('AT\_G <= 25.38', 0.45660969538955837),  
 ('9.70 < E\_G <= 13.62', -0.19823359374058647)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.9694712330882354),  
 (1, -0.45660969538955837),  
 (0, 0.19823359374058647)],  
 1: [(2, 0.9694712330882354),  
 (1, 0.45660969538955837),  
 (0, -0.19823359374058647)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.23714795]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.4046384308538

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr4GG\_Whole.html')

##############################################################################

########### Fitting the model on the training dataset using the best parameters  
model = SVR()  
svr\_param\_grid = {'kernel': ['rbf'],  
 'epsilon' : [0.1],  
 'C' : [300]  
 }  
svr\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
svr4 = GridSearchCV(model, svr\_param\_grid, cv = svr\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1 )  
svr4.fit(X\_G\_train, y\_G\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=SVR(), n\_jobs=-1,  
 param\_grid={'C': [300], 'epsilon': [0.1], 'kernel': ['rbf']},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The svr best score is:', svr4.best\_score\_)

The svr best score is: -0.19855179119854646

# Evaluation of the performance of the model on the training dataset:  
svr\_y\_predtr = svr4.predict(X\_G\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, svr\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, svr\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, svr\_y\_predtr))

The training MAE is: 0.1431573716506935  
The training MSE is: 0.04007184182489732  
The training R2 Score is: 0.8685721593100835

# Evaluation of the performance of the model on the testing dataset:  
svr\_y\_predts = svr4.predict(X\_G\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, svr\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, svr\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, svr\_y\_predts))

The testing MAE is: 0.42273175840012267  
The testing MSE is: 0.5925915346019759  
The testing R2 Score is: -1.160912370469278

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_svr4tr = cross\_val\_score(svr4, X\_G\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_svr4tr)))

The mean training CV score is: 0.1996797379769108

############# On the testing dataset   
score\_svr4ts = cross\_val\_score(svr4, X\_G\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_svr4ts)))

The mean testing CV score is: 0.3074789254556177

############### LIME #################

###### Initializing the explainer using the training dataset  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, svr4.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 280.224553991989  
Prediction\_local [279.76692618]  
Right: 279.8804568737592  
Intercept 280.01028102440023  
Prediction\_local [280.12266403]  
Right: 279.55988692811854  
Intercept 279.9030144837902  
Prediction\_local [280.86235884]  
Right: 280.426986650299  
Intercept 280.3052952246309  
Prediction\_local [279.80939325]  
Right: 279.0206357850338  
Intercept 280.28341995226305  
Prediction\_local [279.8208394]  
Right: 279.03991797107693  
Intercept 279.8976199734231  
Prediction\_local [281.0184033]  
Right: 280.5000053013231  
Intercept 280.30514796468043  
Prediction\_local [280.08987778]  
Right: 278.9390323063231  
Intercept 280.6216183247154  
Prediction\_local [279.5190528]  
Right: 279.30035941845244  
Intercept 280.3939207020241  
Prediction\_local [279.68775904]  
Right: 279.2800568943303  
Intercept 280.28808610313774  
Prediction\_local [279.36462829]  
Right: 279.1999791265987  
Intercept 280.1972519405581  
Prediction\_local [280.24092217]  
Right: 280.17984200652984  
Intercept 280.3478533185382  
Prediction\_local [279.72793777]  
Right: 279.53025338566556  
Intercept 280.4371962480025  
Prediction\_local [279.90688051]  
Right: 279.9226267014495  
Intercept 280.21919712663276  
Prediction\_local [280.0113579]  
Right: 279.25024623053207  
Intercept 279.8055484045778  
Prediction\_local [280.91963125]  
Right: 280.38796484942446  
Intercept 280.2109814206538  
Prediction\_local [279.46877748]  
Right: 279.23001800673575  
Intercept 280.41051267307927  
Prediction\_local [279.95364923]  
Right: 279.98985953477427  
Intercept 280.49937082480807  
Prediction\_local [280.3336353]  
Right: 280.20012173897027  
Intercept 280.4445678743148  
Prediction\_local [279.18648469]  
Right: 278.9903425920588  
Intercept 280.30818438561624  
Prediction\_local [279.75280539]  
Right: 278.84986307898384  
Intercept 280.2329773075908  
Prediction\_local [280.12601341]  
Right: 279.03005654754406  
Intercept 279.7001280023583  
Prediction\_local [281.36117291]  
Right: 280.32531471073037  
Intercept 280.00448517289766  
Prediction\_local [280.01440365]  
Right: 279.91979614052127  
Intercept 280.0933641739483  
Prediction\_local [279.78535845]  
Right: 279.450223481125  
Intercept 280.05095960257256  
Prediction\_local [279.90941349]  
Right: 279.80977390404547  
Intercept 280.07097698002207  
Prediction\_local [280.91438439]  
Right: 280.2687392076114  
Intercept 280.40125598375374  
Prediction\_local [279.46303851]  
Right: 279.31214753937314  
Intercept 280.23120354559285  
Prediction\_local [281.08976121]  
Right: 280.140117434969  
Intercept 280.15632232076143  
Prediction\_local [279.63075398]  
Right: 278.8931720292164  
Intercept 280.1557235754016  
Prediction\_local [281.11501575]  
Right: 280.0629824779121  
Intercept 280.24508611447027  
Prediction\_local [279.47117898]  
Right: 279.63011787089926  
Intercept 279.9432174054344  
Prediction\_local [281.14432655]  
Right: 280.4905603740214  
Intercept 279.8381798770037  
Prediction\_local [280.93093017]  
Right: 279.83074944121296  
Intercept 280.34094602982384  
Prediction\_local [279.77784284]  
Right: 280.48951913126996  
Intercept 279.8551962818734  
Prediction\_local [281.26885723]  
Right: 280.34400866053386  
Intercept 280.40313427345313  
Prediction\_local [279.63377121]  
Right: 279.68299316658204  
Intercept 279.7634018121398  
Prediction\_local [281.33816653]  
Right: 280.0596546546238  
Intercept 280.3676028988898  
Prediction\_local [279.78192929]  
Right: 279.9676813664159  
Intercept 280.4866347516816  
Prediction\_local [279.93581057]  
Right: 279.530177741965  
Intercept 279.9139519936099  
Prediction\_local [281.37699808]  
Right: 280.2096403056195  
Intercept 280.03104579386167  
Prediction\_local [279.70680999]  
Right: 280.45953145126447  
Intercept 280.3499488700268  
Prediction\_local [280.12519624]  
Right: 279.60007520073185  
Intercept 280.18716246000037  
Prediction\_local [279.83719618]  
Right: 279.871284992008  
Intercept 280.4988418614535  
Prediction\_local [279.62423444]  
Right: 280.46612496221184  
Intercept 280.7857717761794  
Prediction\_local [279.33336391]  
Right: 279.2773181405931  
Intercept 280.2610600262093  
Prediction\_local [280.11317495]  
Right: 279.0704371448673  
Intercept 280.193832602954  
Prediction\_local [279.50976984]  
Right: 280.0080627057164  
Intercept 279.94287135833315  
Prediction\_local [279.86106564]  
Right: 279.6398298735712  
Intercept 280.00222958771195  
Prediction\_local [281.0091404]  
Right: 280.1301428978451  
Intercept 280.3921422661488  
Prediction\_local [279.54637437]  
Right: 279.09028025473566  
Intercept 280.23004602020274  
Prediction\_local [279.93530868]  
Right: 279.5046216084999  
Intercept 279.9143228197805  
Prediction\_local [281.47473041]  
Right: 280.5000937317485  
Intercept 279.95765166921126  
Prediction\_local [281.14917561]  
Right: 280.520239071486  
Intercept 280.6350846707313  
Prediction\_local [279.50838496]  
Right: 279.3122715583491  
Intercept 280.1727452017774  
Prediction\_local [279.98613468]  
Right: 280.199635428562  
Intercept 280.1580685843631  
Prediction\_local [279.87321755]  
Right: 279.87920056485456  
Intercept 280.06327080895244  
Prediction\_local [280.11480874]  
Right: 279.6298381805644  
Intercept 280.11492884346484  
Prediction\_local [279.96877964]  
Right: 279.77105659818466  
Intercept 280.4434962562733  
Prediction\_local [280.43322359]  
Right: 279.17991445543896  
Intercept 280.25293946835325  
Prediction\_local [279.79092737]  
Right: 279.6570385495553  
Intercept 280.7462923310727  
Prediction\_local [279.37824076]  
Right: 279.39021762796517  
Intercept 280.3893923801778  
Prediction\_local [279.69566078]  
Right: 279.3425414280733  
Intercept 280.39844547955045  
Prediction\_local [279.71294921]  
Right: 279.02172876915927  
Intercept 280.00253986525314  
Prediction\_local [281.03418998]  
Right: 279.98665239282695  
Intercept 280.1075345117609  
Prediction\_local [279.92455941]  
Right: 279.8901722085341  
Intercept 280.1700154432943  
Prediction\_local [280.01321897]  
Right: 279.42982456534224  
Intercept 279.87391323792957  
Prediction\_local [280.91673615]  
Right: 280.13465714032213  
Intercept 280.26101126587406  
Prediction\_local [280.2412667]  
Right: 279.1096465168041  
Intercept 280.4641821865638  
Prediction\_local [279.73137578]  
Right: 279.0922976556265  
Intercept 280.4677615148689  
Prediction\_local [279.32865965]  
Right: 278.873327124711  
Intercept 280.6594387753684  
Prediction\_local [279.64613955]  
Right: 279.4634591594502  
Intercept 280.1981431935834  
Prediction\_local [279.59172523]  
Right: 280.39640516476004  
Intercept 280.07409309045136  
Prediction\_local [279.93045402]  
Right: 279.3155052944592  
Intercept 280.34789629481884  
Prediction\_local [279.03782778]  
Right: 278.94870122739917  
Intercept 280.20742512498566  
Prediction\_local [279.67913158]  
Right: 279.6196699625166  
Intercept 279.8430140441908  
Prediction\_local [281.19663373]  
Right: 280.3678873674327  
Intercept 279.90882913224874  
Prediction\_local [281.17370634]  
Right: 280.1082587274244  
Intercept 280.20141799290565  
Prediction\_local [279.67495655]  
Right: 280.17099967337873  
Intercept 280.8494516103042  
Prediction\_local [279.46735807]  
Right: 279.9701923147596  
Intercept 279.7450673337216  
Prediction\_local [280.97901858]  
Right: 280.379498274619  
Intercept 280.0526596377607  
Prediction\_local [281.0435787]  
Right: 279.730288992201

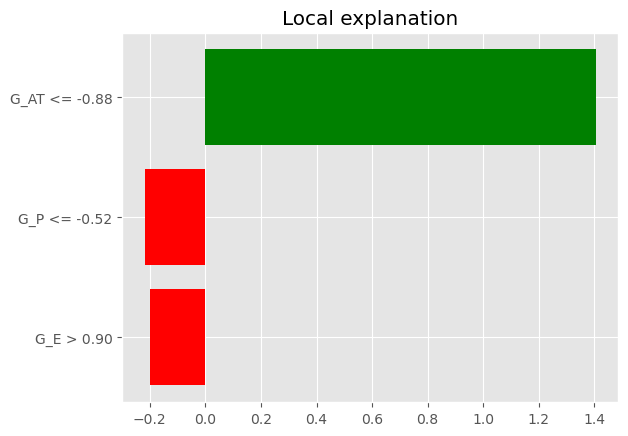
exp

<lime.explanation.Explanation at 0x25d90cadeb0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= -0.88', 1.4059833523962648),  
 ('G\_P <= -0.52', -0.21737858334512175),  
 ('G\_E > 0.90', -0.1976857064995738)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -1.4059833523962648),  
 (2, 0.21737858334512175),  
 (0, 0.1976857064995738)],  
 1: [(1, 1.4059833523962648),  
 (2, -0.21737858334512175),  
 (0, -0.1976857064995738)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.0435787]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.730288992201

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr4GG\_train.html')

################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, svr4.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.5422128998926  
Prediction\_local [281.05773565]  
Right: 280.2590994395691  
Intercept 279.89548386039917  
Prediction\_local [281.02279757]  
Right: 280.25560319389893  
Intercept 280.56283363909114  
Prediction\_local [279.79554667]  
Right: 279.31761307953684  
Intercept 280.3791351687487  
Prediction\_local [279.87050638]  
Right: 279.81798424987505  
Intercept 280.25126286144376  
Prediction\_local [279.2794927]  
Right: 279.5233745577569  
Intercept 280.0949817599727  
Prediction\_local [280.19063252]  
Right: 279.65008324013723  
Intercept 280.44197047152915  
Prediction\_local [279.49156496]  
Right: 278.6957941710821  
Intercept 280.0912078558323  
Prediction\_local [281.18582736]  
Right: 280.33585145016747  
Intercept 280.229271715878  
Prediction\_local [279.5635061]  
Right: 280.6290775270127  
Intercept 279.73837540223735  
Prediction\_local [280.75895747]  
Right: 280.13738717689273  
Intercept 280.4154893216253  
Prediction\_local [279.95870162]  
Right: 279.4397741796793  
Intercept 280.47887915372746  
Prediction\_local [280.15203466]  
Right: 280.5519945865139  
Intercept 280.02789574389413  
Prediction\_local [280.66873575]  
Right: 280.21648608017733  
Intercept 280.64045545728425  
Prediction\_local [280.18346798]  
Right: 280.49721764102264  
Intercept 279.9106505818287  
Prediction\_local [280.09634038]  
Right: 279.80679180395384  
Intercept 280.15666924373255  
Prediction\_local [279.95353867]  
Right: 279.2567663337199  
Intercept 280.4006908369812  
Prediction\_local [279.38025913]  
Right: 280.2208432337051  
Intercept 280.42003172276884  
Prediction\_local [279.66284071]  
Right: 280.0681447725898  
Intercept 280.34747247891585  
Prediction\_local [280.42748789]  
Right: 280.32512680221475  
Intercept 280.2649324854848  
Prediction\_local [280.0527771]  
Right: 279.0524919749615  
Intercept 280.30570146919405  
Prediction\_local [279.85455285]  
Right: 280.41406919430733  
Intercept 280.56444293595786  
Prediction\_local [279.47079949]  
Right: 279.1397381357678  
Intercept 280.0193792680445  
Prediction\_local [280.72059941]  
Right: 280.66321475028985  
Intercept 280.08672020323104  
Prediction\_local [280.58256694]  
Right: 280.42824733747346  
Intercept 280.23076229665145  
Prediction\_local [281.06583218]  
Right: 280.45776124618243  
Intercept 280.02443224447217  
Prediction\_local [279.96143519]  
Right: 279.51564559168247  
Intercept 279.985150444006  
Prediction\_local [281.39275874]  
Right: 282.70526614045656

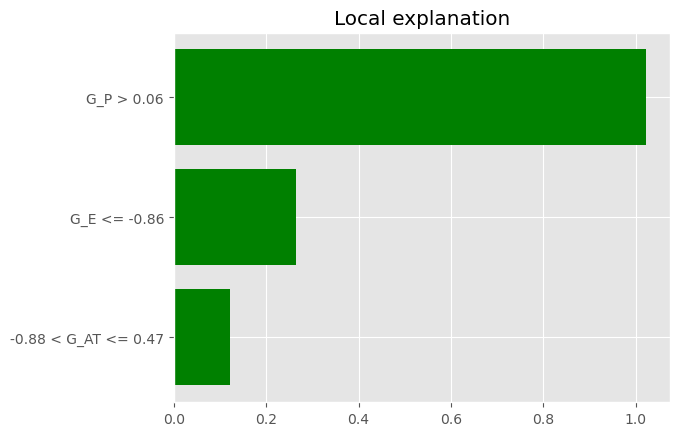
exp

<lime.explanation.Explanation at 0x25d90bc77c0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_P > 0.06', 1.0223851344967263),  
 ('G\_E <= -0.86', 0.26331937243462084),  
 ('-0.88 < G\_AT <= 0.47', 0.1219037913556419)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -1.0223851344967263),  
 (0, -0.26331937243462084),  
 (1, -0.1219037913556419)],  
 1: [(2, 1.0223851344967263),  
 (0, 0.26331937243462084),  
 (1, 0.1219037913556419)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.39275874]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 282.70526614045656

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_svr4GG\_test.html')

##################### END ####################################

#######################################################################################################

#######################################################################################################

#######################################################################################################

#######################################################################################################

##################### RANDOM FOREST REGRESSION ###################

from sklearn.ensemble import RandomForestRegressor

############ MODEL 1: RS Features (X\_R) and RS Target (LL\_R)###############

########### On the whole dataset  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [1, 50, 100, 150, 200],  
 'max\_depth' : ['None', 3, 6, 9, 12, 15],  
 'min\_samples\_leaf' : [1, 2, 3, 4, 5],  
 'min\_samples\_split' : [2, 4, 6, 8, 10, 12],  
 'max\_leaf\_nodes' : ['None', 1, 2, 3, 4, 5],  
 }  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf1 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
rf1.fit(X\_R, y\_R)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': ['None', 3, 6, 9, 12, 15],  
 'max\_leaf\_nodes': ['None', 1, 2, 3, 4, 5],  
 'min\_samples\_leaf': [1, 2, 3, 4, 5],  
 'min\_samples\_split': [2, 4, 6, 8, 10, 12],  
 'n\_estimators': [51, 50, 100, 150, 200]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The rf best parameters are:', rf1.best\_params\_)  
print('The rf best score is:', rf1.best\_score\_)

The rf best parameters are: {'max\_depth': 12, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 50}  
The rf best score is: -0.1192655566038503

#The rf best parameters are: {'max\_depth': 12, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 50}  
#The rf best score is: -0.1192655566038503

# Evaluation of the performance of the model on the whole dataset:  
rf\_y\_pred = rf1.predict(X\_R)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_R, rf\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_R, rf\_y\_pred))  
print('R2 Score:', r2\_score(y\_R, rf\_y\_pred))

Mean Absolute Error: 0.22626307195245154  
Mean Squared Error: 0.08158695814592434  
R2 Score: 0.6002193767056871

################ LIME ################

###### Initializing the explainer on the whole features dataset  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the full datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, rf1.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 281.3743358299131  
Prediction\_local [281.57223371]  
Right: 281.7685741831564  
Intercept 281.3042569212885  
Prediction\_local [281.7075895]  
Right: 281.58328287293796  
Intercept 281.4220223110887  
Prediction\_local [281.42618799]  
Right: 281.32625318934043  
Intercept 281.46180394175263  
Prediction\_local [281.27090045]  
Right: 281.18983175426814  
Intercept 281.5584711478571  
Prediction\_local [281.15259076]  
Right: 281.11065161189634  
Intercept 281.4599113029925  
Prediction\_local [281.30658204]  
Right: 281.302260115196  
Intercept 281.4798020030782  
Prediction\_local [281.34511774]  
Right: 281.2726455430724  
Intercept 281.4711745018018  
Prediction\_local [281.3673874]  
Right: 281.2862522097391  
Intercept 281.4117812643002  
Prediction\_local [281.49254404]  
Right: 281.5637545242731  
Intercept 281.40208042672543  
Prediction\_local [281.60978681]  
Right: 281.6560187065663  
Intercept 281.3076967709891  
Prediction\_local [281.8441895]  
Right: 281.86936120274066  
Intercept 281.3168649069268  
Prediction\_local [281.82536824]  
Right: 281.8516448136949  
Intercept 281.46104934369674  
Prediction\_local [281.56108965]  
Right: 281.6270452574907  
Intercept 281.29575664113406  
Prediction\_local [281.72569318]  
Right: 281.59631311314604  
Intercept 281.4597122668637  
Prediction\_local [281.39083127]  
Right: 281.2977554382436  
Intercept 281.5677317724666  
Prediction\_local [281.06301447]  
Right: 281.19383238918874  
Intercept 281.496948892514  
Prediction\_local [281.40773941]  
Right: 281.1070088738729  
Intercept 281.4866358228712  
Prediction\_local [281.30465166]  
Right: 281.1236307786348  
Intercept 281.4697574032502  
Prediction\_local [281.36382174]  
Right: 281.3165272383796  
Intercept 281.4143855853642  
Prediction\_local [281.43349937]  
Right: 281.2862522097391  
Intercept 281.3920351007541  
Prediction\_local [281.5491242]  
Right: 281.5740407742731  
Intercept 281.3971124144636  
Prediction\_local [281.46900093]  
Right: 281.59808970162356  
Intercept 281.3065076888986  
Prediction\_local [281.83133379]  
Right: 281.878334536074  
Intercept 281.29472877547806  
Prediction\_local [281.86849744]  
Right: 281.861728904604  
Intercept 281.41666866855286  
Prediction\_local [281.58361635]  
Right: 281.6515531677471  
Intercept 281.3691182702377  
Prediction\_local [281.72404863]  
Right: 281.58328287293796  
Intercept 281.43135999600315  
Prediction\_local [281.30660491]  
Right: 281.30658699153827  
Intercept 281.4590169433968  
Prediction\_local [281.41773261]  
Right: 281.3158273830028  
Intercept 281.53062299345487  
Prediction\_local [281.1286421]  
Right: 281.1312168000784  
Intercept 281.41460081094584  
Prediction\_local [281.36295682]  
Right: 281.1755673043261  
Intercept 281.4600155069766  
Prediction\_local [281.32794699]  
Right: 281.24291218274504  
Intercept 281.4742622992378  
Prediction\_local [281.3615821]  
Right: 281.2631015528643  
Intercept 281.39796914625475  
Prediction\_local [281.41259427]  
Right: 281.503286510175  
Intercept 281.4309640155607  
Prediction\_local [281.34656376]  
Right: 281.5136840446202  
Intercept 281.2900374573494  
Prediction\_local [281.82378396]  
Right: 281.878334536074  
Intercept 281.3088011887278  
Prediction\_local [281.81829252]  
Right: 281.8516448136949  
Intercept 281.3732294980115  
Prediction\_local [281.59745356]  
Right: 281.7685741831564  
Intercept 281.33315446255824  
Prediction\_local [281.76535916]  
Right: 281.6208210234024  
Intercept 281.53609170225883  
Prediction\_local [281.20768302]  
Right: 281.22228217902364  
Intercept 281.5614186000582  
Prediction\_local [281.17322413]  
Right: 281.0571974161777  
Intercept 281.59407858258356  
Prediction\_local [281.11370476]  
Right: 280.93199996707  
Intercept 281.5783858646316  
Prediction\_local [280.96564821]  
Right: 280.9319738958931  
Intercept 281.4900490046242  
Prediction\_local [281.13523855]  
Right: 281.06992419632303  
Intercept 281.5008190738178  
Prediction\_local [281.31815032]  
Right: 281.2799070701057  
Intercept 281.5443930591289  
Prediction\_local [281.18066825]  
Right: 281.2282801967976  
Intercept 281.42948137223175  
Prediction\_local [281.32997114]  
Right: 281.3166112728916  
Intercept 281.3255283970832  
Prediction\_local [281.8079994]  
Right: 281.8653042958659  
Intercept 281.29264049239777  
Prediction\_local [281.80668506]  
Right: 281.8516448136949  
Intercept 281.3244667142054  
Prediction\_local [281.67771876]  
Right: 281.6270452574907  
Intercept 281.28278539637375  
Prediction\_local [281.67820879]  
Right: 281.6208210234024  
Intercept 281.43127271126446  
Prediction\_local [281.52585119]  
Right: 281.4435531888853  
Intercept 281.50468498656414  
Prediction\_local [281.1235851]  
Right: 281.15787354090475  
Intercept 281.52095338142414  
Prediction\_local [281.13806899]  
Right: 280.924811819556  
Intercept 281.57759896200355  
Prediction\_local [281.06411699]  
Right: 281.0192023748926  
Intercept 281.5221963715167  
Prediction\_local [281.04837875]  
Right: 281.0394040544732  
Intercept 281.5259535505162  
Prediction\_local [281.09094928]  
Right: 281.14398779039783  
Intercept 281.54569130912614  
Prediction\_local [281.05574676]  
Right: 281.129339855255  
Intercept 281.548220940416  
Prediction\_local [281.14708308]  
Right: 281.4213336618592  
Intercept 281.299301272019  
Prediction\_local [281.83634485]  
Right: 281.8653042958659  
Intercept 281.3096349015103  
Prediction\_local [281.75792341]  
Right: 281.836985955528  
Intercept 281.44339684487636  
Prediction\_local [281.4795201]  
Right: 281.6270452574907  
Intercept 281.349967480097  
Prediction\_local [281.68826791]  
Right: 281.58328287293796  
Intercept 281.47031419852453  
Prediction\_local [281.27451587]  
Right: 281.25550911493525  
Intercept 281.48002981102496  
Prediction\_local [281.23904816]  
Right: 281.10249587436095  
Intercept 281.5694802366394  
Prediction\_local [281.02723815]  
Right: 280.90952854761224  
Intercept 281.5499619928523  
Prediction\_local [280.98656566]  
Right: 280.9243593610094  
Intercept 281.49936972431135  
Prediction\_local [281.23168559]  
Right: 281.0716721328309  
Intercept 281.47616854778425  
Prediction\_local [281.36340353]  
Right: 281.2632316509113  
Intercept 281.53280282937214  
Prediction\_local [281.20135261]  
Right: 281.25547655905973  
Intercept 281.5059029364373  
Prediction\_local [281.12865545]  
Right: 281.30844842964876  
Intercept 281.29032828205874  
Prediction\_local [281.83384986]  
Right: 281.8653042958659  
Intercept 281.2960795469067  
Prediction\_local [281.83200356]  
Right: 281.8516448136949  
Intercept 281.43743753760293  
Prediction\_local [281.54515138]  
Right: 281.6270452574907  
Intercept 281.31845760809045  
Prediction\_local [281.71268722]  
Right: 281.59631311314604  
Intercept 281.45143262911364  
Prediction\_local [281.37786761]  
Right: 281.34483315727823  
Intercept 281.57638610276115  
Prediction\_local [281.0944881]  
Right: 281.19501176103705  
Intercept 281.5178058272818  
Prediction\_local [281.08022086]  
Right: 280.90952854761224  
Intercept 281.5421230534617  
Prediction\_local [280.99281386]  
Right: 280.9430435928628  
Intercept 281.5241104766376  
Prediction\_local [281.25685637]  
Right: 281.11939451542895  
Intercept 281.4948457230681  
Prediction\_local [281.35691805]  
Right: 281.2802305785334  
Intercept 281.43471780089544  
Prediction\_local [281.35122974]  
Right: 281.428898329613  
Intercept 281.49458575008157  
Prediction\_local [281.22619694]  
Right: 281.4953684506031  
Intercept 281.36559839003223  
Prediction\_local [281.68530427]  
Right: 281.7820776017914  
Intercept 281.25480027149024  
Prediction\_local [281.8395248]  
Right: 281.86936120274066  
Intercept 281.34168562298794  
Prediction\_local [281.56251111]  
Right: 281.7685741831564  
Intercept 281.3395338009029  
Prediction\_local [281.74197989]  
Right: 281.59631311314604  
Intercept 281.40260271106627  
Prediction\_local [281.42732067]  
Right: 281.4959042310155  
Intercept 281.5036980665337  
Prediction\_local [281.27167149]  
Right: 281.2063737576954  
Intercept 281.57626693991307  
Prediction\_local [281.09169619]  
Right: 281.00619699687167  
Intercept 281.53805368415334  
Prediction\_local [281.05841686]  
Right: 281.02440222860207  
Intercept 281.54208591686853  
Prediction\_local [281.19173474]  
Right: 281.17308087053993  
Intercept 281.4641142966404  
Prediction\_local [281.33302991]  
Right: 281.2802305785334  
Intercept 281.4808721868262  
Prediction\_local [281.32710547]  
Right: 281.3426672383796  
Intercept 281.41501527624956  
Prediction\_local [281.39792897]  
Right: 281.4564788599784  
Intercept 281.2924409654158  
Prediction\_local [281.839934]  
Right: 281.878334536074  
Intercept 281.2965887054868  
Prediction\_local [281.86797951]  
Right: 281.8516448136949  
Intercept 281.40023438731885  
Prediction\_local [281.54981417]  
Right: 281.6605789847897  
Intercept 281.3175392050719  
Prediction\_local [281.72223238]  
Right: 281.63607107453333  
Intercept 281.318255695683  
Prediction\_local [281.65662444]  
Right: 281.83036610834205  
Intercept 281.47499504029474  
Prediction\_local [281.2248024]  
Right: 281.39715640634324  
Intercept 281.45297382392624  
Prediction\_local [281.28441247]  
Right: 281.09147441859386  
Intercept 281.45880931463324  
Prediction\_local [281.28493213]  
Right: 281.27127200348235  
Intercept 281.4405386558912  
Prediction\_local [281.43956759]  
Right: 281.5246582540513  
Intercept 281.42461165473026  
Prediction\_local [281.58631707]  
Right: 281.6190542296743  
Intercept 281.3944180787571  
Prediction\_local [281.4314797]  
Right: 281.67132668748576  
Intercept 281.4392859774071  
Prediction\_local [281.5462145]  
Right: 281.5711005506727  
Intercept 281.33616943118477  
Prediction\_local [281.79981167]  
Right: 281.8377228395248  
Intercept 281.3311138761243  
Prediction\_local [281.64920974]  
Right: 281.86936120274066

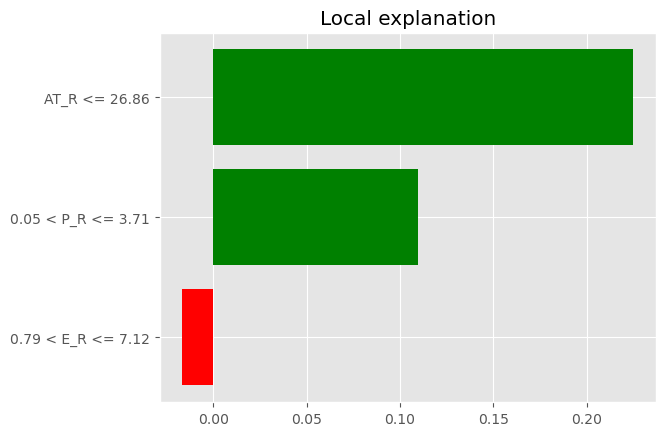
exp

<lime.explanation.Explanation at 0x1c3c0eda3d0>

##### Showing the explanable table  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('AT\_R <= 26.86', 0.22494548496307262),  
 ('0.05 < P\_R <= 3.71', 0.10983278846253618),  
 ('0.79 < E\_R <= 7.12', -0.016682409687601565)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.22494548496307262),  
 (2, -0.10983278846253618),  
 (0, 0.016682409687601565)],  
 1: [(1, 0.22494548496307262),  
 (2, 0.10983278846253618),  
 (0, -0.016682409687601565)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.64920974]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.86936120274066

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf1RR\_Whole.html')

####################################################################################################

#The rf best parameters are: {'max\_depth': 12, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 50}

########### Fitting the model on the training dataset using the best parameters  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [50],  
 'max\_depth' : [12],  
 'min\_samples\_leaf' : [4],  
 'min\_samples\_split' : [2],  
 'max\_leaf\_nodes' : [5],  
 }  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf1 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
rf1.fit(X\_R\_train, y\_R\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': [12], 'max\_leaf\_nodes': [5],  
 'min\_samples\_leaf': [4], 'min\_samples\_split': [2],  
 'n\_estimators': [50]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The rf best score is:', rf1.best\_score\_)

The rf best score is: -0.12733654942614586

# Evaluation of the performance of the model on the training dataset:  
rf\_y\_predtr = rf1.predict(X\_R\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_R\_train, rf\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_R\_train, rf\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_R\_train, rf\_y\_predtr))

The training MAE is: 0.22189622962062255  
The training MSE is: 0.07815259951357852  
The training R2 Score is: 0.6300693261022373

# Evaluation of the performance of the model on the testing dataset:  
rf\_y\_predts = rf1.predict(X\_R\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_R\_test, rf\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_R\_test, rf\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_R\_test, rf\_y\_predts))

The testing MAE is: 0.24595332965104802  
The testing MSE is: 0.097193088142351  
The testing R2 Score is: 0.3308309414923778

########### Cross-validation ###########################

############# On the training dataset  
   
score\_rf1tr = cross\_val\_score(rf1, X\_R\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_rf1tr)))

The mean training CV score is: 0.1331366152542438

############# On the testing dataset   
score\_rf1ts = cross\_val\_score(rf1, X\_R\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_rf1ts)))

The mean testing CV score is: 0.11360176276103076

################ LIME ################

###### Initializing the explainer on the training dataset  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, rf1.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.392159721525  
Prediction\_local [281.3067527]  
Right: 281.24422182546107  
Intercept 281.4722549629452  
Prediction\_local [281.15240023]  
Right: 281.2332394470505  
Intercept 281.2579131466529  
Prediction\_local [281.68655747]  
Right: 281.80763922425234  
Intercept 281.42328045630546  
Prediction\_local [281.15606257]  
Right: 281.08918988802327  
Intercept 281.41151161573043  
Prediction\_local [281.23434301]  
Right: 281.1543178955272  
Intercept 281.2371674349878  
Prediction\_local [281.81402128]  
Right: 281.8599514670312  
Intercept 281.44549461094  
Prediction\_local [281.16132498]  
Right: 281.1199028450668  
Intercept 281.39891778687587  
Prediction\_local [281.09488836]  
Right: 281.18223277186576  
Intercept 281.4280366617869  
Prediction\_local [281.07979146]  
Right: 280.8647824392362  
Intercept 281.3710087542315  
Prediction\_local [281.19857472]  
Right: 281.094831299788  
Intercept 281.415263177881  
Prediction\_local [281.1959364]  
Right: 281.5223255779563  
Intercept 281.4814355838617  
Prediction\_local [281.10046457]  
Right: 280.94106343214355  
Intercept 281.3831177743585  
Prediction\_local [281.39161954]  
Right: 281.43163569678876  
Intercept 281.34933515789174  
Prediction\_local [281.27904073]  
Right: 281.4945383834519  
Intercept 281.2472247159005  
Prediction\_local [281.7115138]  
Right: 281.8227820946258  
Intercept 281.4627784149668  
Prediction\_local [281.17184908]  
Right: 280.90367429435923  
Intercept 281.2305356904757  
Prediction\_local [281.76399403]  
Right: 281.8599514670312  
Intercept 281.35077947116974  
Prediction\_local [281.41662989]  
Right: 281.4059770106364  
Intercept 281.41338774269224  
Prediction\_local [281.33972368]  
Right: 281.28789196623086  
Intercept 281.3968428468089  
Prediction\_local [281.29144523]  
Right: 281.2071482801334  
Intercept 281.3889970075426  
Prediction\_local [281.28973396]  
Right: 281.1604169431462  
Intercept 281.2499613492744  
Prediction\_local [281.74814835]  
Right: 281.84396182600557  
Intercept 281.4007001255771  
Prediction\_local [281.36301179]  
Right: 281.21940149118177  
Intercept 281.4027595878973  
Prediction\_local [281.16739296]  
Right: 281.1320409819972  
Intercept 281.3939224615227  
Prediction\_local [281.20521948]  
Right: 281.21187006261033  
Intercept 281.32693306816213  
Prediction\_local [281.41872327]  
Right: 281.607170890218  
Intercept 281.3476527144577  
Prediction\_local [281.22888756]  
Right: 281.32354849393084  
Intercept 281.35611816649214  
Prediction\_local [281.39573938]  
Right: 281.5828085638705  
Intercept 281.49184726587106  
Prediction\_local [281.05305077]  
Right: 281.0657116744767  
Intercept 281.2675070642876  
Prediction\_local [281.68731011]  
Right: 281.5663376105773  
Intercept 281.44415090076524  
Prediction\_local [281.07645574]  
Right: 280.88995055528784  
Intercept 281.2090949373222  
Prediction\_local [281.76155942]  
Right: 281.8599514670312  
Intercept 281.3800402676141  
Prediction\_local [281.50963475]  
Right: 281.7460399491483  
Intercept 281.22987485994537  
Prediction\_local [281.7602157]  
Right: 281.8599514670312  
Intercept 281.14433526447885  
Prediction\_local [281.8370665]  
Right: 281.8227820946258  
Intercept 281.4791626043876  
Prediction\_local [281.04693286]  
Right: 281.26779692587706  
Intercept 281.2191838100631  
Prediction\_local [281.67797612]  
Right: 281.558145302885  
Intercept 281.23734819436925  
Prediction\_local [281.55487856]  
Right: 281.5662409492322  
Intercept 281.52281640293563  
Prediction\_local [281.05483787]  
Right: 280.9300712300362  
Intercept 281.4028962304634  
Prediction\_local [281.23365734]  
Right: 281.27065327164496  
Intercept 281.24787218934307  
Prediction\_local [281.73281582]  
Right: 281.81145790010163  
Intercept 281.3926595105412  
Prediction\_local [281.43840012]  
Right: 281.408164911041  
Intercept 281.3763871368602  
Prediction\_local [281.35421073]  
Right: 281.34102089126696  
Intercept 281.2388689100955  
Prediction\_local [281.7710163]  
Right: 281.8599514670312  
Intercept 281.4378736160409  
Prediction\_local [281.14620686]  
Right: 281.1529636899559  
Intercept 281.41278219443024  
Prediction\_local [281.24451563]  
Right: 281.094831299788  
Intercept 281.3986848837881  
Prediction\_local [281.38856728]  
Right: 281.5461064993127  
Intercept 281.3706518040515  
Prediction\_local [281.37075217]  
Right: 281.30330359528847  
Intercept 281.2801146936993  
Prediction\_local [281.64355711]  
Right: 281.5663376105773  
Intercept 281.4806053943278  
Prediction\_local [281.25650781]  
Right: 281.12947493886423  
Intercept 281.44168425882526  
Prediction\_local [281.15164616]  
Right: 281.1811720760796  
Intercept 281.2771875374295  
Prediction\_local [281.69837356]  
Right: 281.8309744023181  
Intercept 281.36191632328166  
Prediction\_local [281.43426112]  
Right: 281.7324728368489  
Intercept 281.48617989270673  
Prediction\_local [281.11412883]  
Right: 280.8647824392362  
Intercept 281.4432373151811  
Prediction\_local [281.33164496]  
Right: 281.3805206401731  
Intercept 281.2725009761307  
Prediction\_local [281.61444902]  
Right: 281.7043303052857  
Intercept 281.37852781273074  
Prediction\_local [281.32466775]  
Right: 281.22873090294644  
Intercept 281.3396468383518  
Prediction\_local [281.1983535]  
Right: 281.1974705047629  
Intercept 281.3634793636263  
Prediction\_local [281.286221]  
Right: 281.16971376854303  
Intercept 281.29893493528124  
Prediction\_local [281.30088169]  
Right: 281.18148636629206  
Intercept 281.5388636493959  
Prediction\_local [281.00627517]  
Right: 280.8647824392362  
Intercept 281.4266180681328  
Prediction\_local [281.18935412]  
Right: 280.9724443805245  
Intercept 281.33919382034577  
Prediction\_local [281.28851848]  
Right: 281.348376622857  
Intercept 281.2635082133745  
Prediction\_local [281.78188421]  
Right: 281.8227820946258  
Intercept 281.3037426836809  
Prediction\_local [281.57415143]  
Right: 281.5663376105773  
Intercept 281.385389862417  
Prediction\_local [281.38469963]  
Right: 281.56978190280506  
Intercept 281.2650635358686  
Prediction\_local [281.68083921]  
Right: 281.5825076292325  
Intercept 281.5151468596448  
Prediction\_local [281.0175314]  
Right: 280.8959687388036  
Intercept 281.3966291330286  
Prediction\_local [281.23994325]  
Right: 281.1604169431462  
Intercept 281.51207240912015  
Prediction\_local [280.94522617]  
Right: 280.8959687388036  
Intercept 281.4572513369275  
Prediction\_local [281.19127464]  
Right: 281.1788592347249  
Intercept 281.22264463983515  
Prediction\_local [281.72926923]  
Right: 281.8599514670312  
Intercept 281.32896414545814  
Prediction\_local [281.30435112]  
Right: 281.1658555329588  
Intercept 281.5001049909102  
Prediction\_local [281.00883099]  
Right: 280.968531824969  
Intercept 281.3539325107577  
Prediction\_local [281.32646173]  
Right: 281.2424613033511  
Intercept 281.2848636810193  
Prediction\_local [281.76645669]  
Right: 281.8437267832705  
Intercept 281.3684000936312  
Prediction\_local [281.42905398]  
Right: 281.5828085638705  
Intercept 281.45226953421917  
Prediction\_local [281.25317077]  
Right: 281.4063467523108  
Intercept 281.4061702564093  
Prediction\_local [281.41738716]  
Right: 281.3773693478192  
Intercept 281.3015132235674  
Prediction\_local [281.76436172]  
Right: 281.8517591593389  
Intercept 281.30341676613284  
Prediction\_local [281.65729031]  
Right: 281.5825076292325

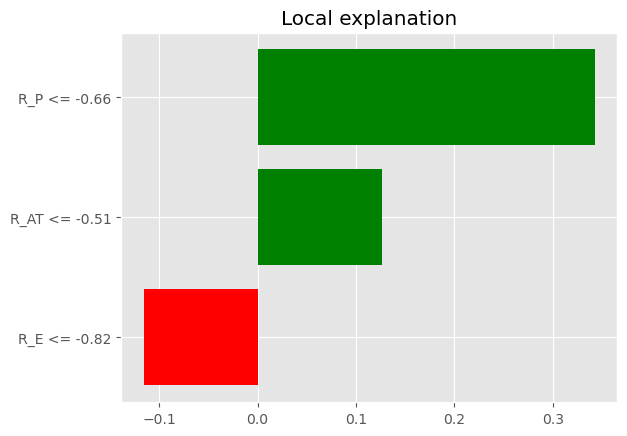
exp

<lime.explanation.Explanation at 0x25d924cc8e0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P <= -0.66', 0.34270837038996677),  
 ('R\_AT <= -0.51', 0.12655009546239707),  
 ('R\_E <= -0.82', -0.11538492480509675)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.34270837038996677),  
 (1, -0.12655009546239707),  
 (0, 0.11538492480509675)],  
 1: [(2, 0.34270837038996677),  
 (1, 0.12655009546239707),  
 (0, -0.11538492480509675)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.65729031]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.5825076292325

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf1RR\_train.html')

################################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, rf1.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 281.30554147008377  
Prediction\_local [281.55827207]  
Right: 281.5663376105773  
Intercept 281.33424097534095  
Prediction\_local [281.44782926]  
Right: 281.7324728368489  
Intercept 281.40029255185516  
Prediction\_local [281.06463829]  
Right: 280.97209805119115  
Intercept 281.3484849405207  
Prediction\_local [281.45231884]  
Right: 281.57370814739375  
Intercept 281.46404619080107  
Prediction\_local [281.05051897]  
Right: 281.2074847694452  
Intercept 281.3368154783082  
Prediction\_local [281.50066245]  
Right: 281.5676090997747  
Intercept 281.3627602790768  
Prediction\_local [281.48669344]  
Right: 281.56368285518596  
Intercept 281.2872546404938  
Prediction\_local [281.40327454]  
Right: 281.5828085638705  
Intercept 281.36419353861424  
Prediction\_local [281.27382313]  
Right: 281.16801807950986  
Intercept 281.2755933776633  
Prediction\_local [281.45621429]  
Right: 281.607170890218  
Intercept 281.5577512234393  
Prediction\_local [281.1211775]  
Right: 280.9018628439125  
Intercept 281.37740927280805  
Prediction\_local [281.44305848]  
Right: 281.41002830836163  
Intercept 281.30831516719417  
Prediction\_local [281.571571]  
Right: 281.558145302885  
Intercept 281.2553540551875  
Prediction\_local [281.65123123]  
Right: 281.6779286617129  
Intercept 281.41863364707893  
Prediction\_local [281.19697916]  
Right: 281.1476549977853  
Intercept 281.38808179186873  
Prediction\_local [281.06193348]  
Right: 280.9018628439125  
Intercept 281.3142655016289  
Prediction\_local [281.36370104]  
Right: 281.5338529589087  
Intercept 281.3384730887029  
Prediction\_local [281.4409331]  
Right: 281.5726350211312  
Intercept 281.39957026251597  
Prediction\_local [281.21663062]  
Right: 281.1604169431462  
Intercept 281.379647297522  
Prediction\_local [281.39777708]  
Right: 281.1604169431462  
Intercept 281.2509733163852  
Prediction\_local [281.75228075]  
Right: 281.8599514670312  
Intercept 281.4866426895104  
Prediction\_local [281.1070194]  
Right: 281.1183796102455  
Intercept 281.32723288102125  
Prediction\_local [281.37776142]  
Right: 281.5828085638705  
Intercept 281.293209006855  
Prediction\_local [281.63942195]  
Right: 281.558145302885  
Intercept 281.37818175663915  
Prediction\_local [281.66471113]  
Right: 281.8309744023181  
Intercept 281.4352028213586  
Prediction\_local [281.13335895]  
Right: 281.1291914099679  
Intercept 281.401146953651  
Prediction\_local [281.22148433]  
Right: 281.16801807950986

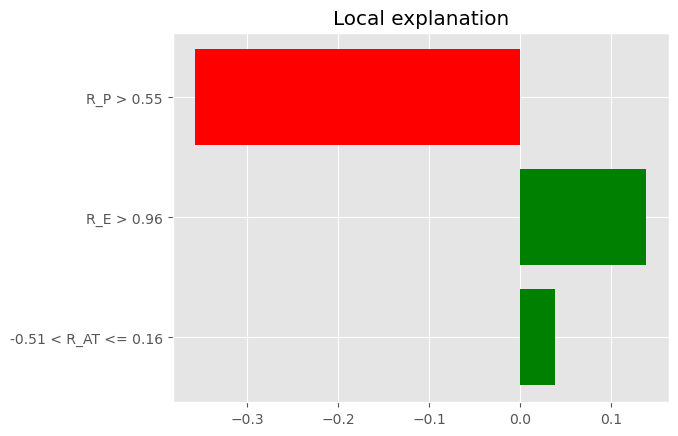
exp

<lime.explanation.Explanation at 0x25d91fc8940>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P > 0.55', -0.3565505602841042),  
 ('R\_E > 0.96', 0.1383340771952285),  
 ('-0.51 < R\_AT <= 0.16', 0.038553863056648256)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 0.3565505602841042),  
 (0, -0.1383340771952285),  
 (1, -0.038553863056648256)],  
 1: [(2, -0.3565505602841042),  
 (0, 0.1383340771952285),  
 (1, 0.038553863056648256)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.22148433]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.16801807950986

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf1RR\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

############ MODEL 2: GT Features (X\_G) and RS Target (LL\_R) ###############

########### On the whole dataset  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [1, 50, 100, 150, 200],  
 'max\_depth' : ['None', 3, 6, 9, 12, 15],  
 'min\_samples\_leaf' : [1, 2, 3, 4, 5],  
 'min\_samples\_split' : [2, 4, 6, 8, 10, 12],  
 'max\_leaf\_nodes' : ['None', 1, 2, 3, 4, 5],  
 }  
  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf2 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
rf2.fit(X\_G, y\_R)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': ['None', 3, 6, 9, 12, 15],  
 'max\_leaf\_nodes': ['None', 1, 2, 3, 4, 5],  
 'min\_samples\_leaf': [1, 2, 3, 4, 5],  
 'min\_samples\_split': [2, 4, 6, 8, 10, 12],  
 'n\_estimators': [1, 50, 100, 150, 200]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The rf best parameters are:', rf2.best\_params\_)  
print('The rf best score is:', rf2.best\_score\_)

The rf best parameters are: {'max\_depth': 6, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 5, 'min\_samples\_split': 8, 'n\_estimators': 1}  
The rf best score is: -0.07295452503009017

#The rf best parameters are: {'max\_depth': 6, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 5, 'min\_samples\_split': 8, 'n\_estimators': 1}  
#The rf best score is: -0.07295452503009017

# Evaluation of the performance of the rf regression model on the whole dataset:  
rf\_y\_pred = rf2.predict(X\_G)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_R, rf\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_R, rf\_y\_pred))  
print('R2 Score:', r2\_score(y\_R, rf\_y\_pred))

Mean Absolute Error: 0.21004173488990704  
Mean Squared Error: 0.07060544335610859  
R2 Score: 0.654029408690653

#####################LIME ##############################

###### Initializing the explainer on the whole datset  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the whole XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, rf2.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 281.2887489332633  
Prediction\_local [281.86308546]  
Right: 281.9489655172413  
Intercept 281.18825677201045  
Prediction\_local [281.90716382]  
Right: 281.4584615384615  
Intercept 281.42537358767675  
Prediction\_local [281.13190069]  
Right: 281.03622222222214  
Intercept 281.4614641520652  
Prediction\_local [281.06270617]  
Right: 281.03622222222214  
Intercept 281.49490487806673  
Prediction\_local [281.12900978]  
Right: 281.03622222222214  
Intercept 281.59692013607184  
Prediction\_local [281.11971531]  
Right: 281.03622222222214  
Intercept 281.5187122112941  
Prediction\_local [281.13810053]  
Right: 281.03622222222214  
Intercept 281.5542748349649  
Prediction\_local [281.14571814]  
Right: 281.03622222222214  
Intercept 281.3457395454719  
Prediction\_local [281.76299274]  
Right: 281.9489655172413  
Intercept 281.2368218291462  
Prediction\_local [281.88238855]  
Right: 281.9489655172413  
Intercept 281.27023268355236  
Prediction\_local [281.8479984]  
Right: 281.9489655172413  
Intercept 281.21887600793974  
Prediction\_local [281.89642705]  
Right: 281.9489655172413  
Intercept 281.2459971504386  
Prediction\_local [281.82444149]  
Right: 281.76916666666665  
Intercept 281.2873627079746  
Prediction\_local [281.79196752]  
Right: 281.76916666666665  
Intercept 281.5301964138024  
Prediction\_local [281.08356711]  
Right: 281.03622222222214  
Intercept 281.55961929711947  
Prediction\_local [281.01198809]  
Right: 281.03622222222214  
Intercept 281.49737300181766  
Prediction\_local [281.12662345]  
Right: 281.03622222222214  
Intercept 281.5544801627192  
Prediction\_local [281.10272947]  
Right: 281.03622222222214  
Intercept 281.570090071381  
Prediction\_local [280.94631537]  
Right: 281.03622222222214  
Intercept 281.54315132093905  
Prediction\_local [281.02952676]  
Right: 281.03622222222214  
Intercept 281.5936162118362  
Prediction\_local [281.17830053]  
Right: 281.03622222222214  
Intercept 281.19844375201643  
Prediction\_local [281.88653992]  
Right: 281.9489655172413  
Intercept 281.17498584948527  
Prediction\_local [281.79452509]  
Right: 281.9489655172413  
Intercept 281.1707378038856  
Prediction\_local [281.83192726]  
Right: 281.9489655172413  
Intercept 281.23345612119033  
Prediction\_local [281.85117412]  
Right: 281.76916666666665  
Intercept 281.319420195695  
Prediction\_local [281.72526004]  
Right: 281.4584615384615  
Intercept 281.4421516586668  
Prediction\_local [281.07274778]  
Right: 281.03622222222214  
Intercept 281.4933596372396  
Prediction\_local [281.04272513]  
Right: 281.03622222222214  
Intercept 281.4965686285329  
Prediction\_local [281.13104978]  
Right: 281.03622222222214  
Intercept 281.4647291409024  
Prediction\_local [281.15502583]  
Right: 281.03622222222214  
Intercept 281.34764621295875  
Prediction\_local [281.6823942]  
Right: 281.5777777777778  
Intercept 281.33254588175225  
Prediction\_local [281.74070962]  
Right: 281.5777777777778  
Intercept 281.43862159461605  
Prediction\_local [281.59330155]  
Right: 281.5777777777778  
Intercept 281.279872329157  
Prediction\_local [281.84926958]  
Right: 281.9489655172413  
Intercept 281.2827100679488  
Prediction\_local [281.85716968]  
Right: 281.4584615384615  
Intercept 281.2608975328099  
Prediction\_local [281.76346499]  
Right: 281.76916666666665  
Intercept 281.24694313068676  
Prediction\_local [281.74084302]  
Right: 281.76916666666665  
Intercept 281.21589562930916  
Prediction\_local [281.77859168]  
Right: 281.76916666666665  
Intercept 281.420633008773  
Prediction\_local [281.21145309]  
Right: 281.03622222222214  
Intercept 281.5246905281538  
Prediction\_local [281.10847258]  
Right: 281.03622222222214  
Intercept 281.5389224951203  
Prediction\_local [281.08243568]  
Right: 281.03622222222214  
Intercept 281.5177128886356  
Prediction\_local [281.089279]  
Right: 281.03622222222214  
Intercept 281.5774989180724  
Prediction\_local [281.05959716]  
Right: 281.03622222222214  
Intercept 281.44499844245536  
Prediction\_local [281.64039432]  
Right: 281.5777777777778  
Intercept 281.2912388455647  
Prediction\_local [281.78088552]  
Right: 281.9489655172413  
Intercept 281.4405782626501  
Prediction\_local [281.28800203]  
Right: 281.9489655172413  
Intercept 281.22222047869394  
Prediction\_local [281.87748839]  
Right: 281.9489655172413  
Intercept 281.20900580764845  
Prediction\_local [281.8638923]  
Right: 281.9489655172413  
Intercept 281.41992659938336  
Prediction\_local [281.62243818]  
Right: 281.4584615384615  
Intercept 281.32004822687634  
Prediction\_local [281.59849683]  
Right: 281.4584615384615  
Intercept 281.3173663589801  
Prediction\_local [281.63233794]  
Right: 281.4584615384615  
Intercept 281.54453100084305  
Prediction\_local [280.95130233]  
Right: 281.03622222222214  
Intercept 281.5207836742591  
Prediction\_local [281.14063435]  
Right: 281.03622222222214  
Intercept 281.52534311182694  
Prediction\_local [281.11075343]  
Right: 281.03622222222214  
Intercept 281.5079770738218  
Prediction\_local [281.15627741]  
Right: 281.03622222222214  
Intercept 281.60296561797327  
Prediction\_local [281.05118695]  
Right: 281.03622222222214  
Intercept 281.4879848599307  
Prediction\_local [281.22178951]  
Right: 281.03622222222214  
Intercept 281.52827593164074  
Prediction\_local [281.27231911]  
Right: 281.9489655172413  
Intercept 281.2637440641583  
Prediction\_local [281.73078081]  
Right: 281.9489655172413  
Intercept 281.2262427381279  
Prediction\_local [281.79034587]  
Right: 281.9489655172413  
Intercept 281.2989530359705  
Prediction\_local [281.72101758]  
Right: 281.76916666666665  
Intercept 281.40366834032443  
Prediction\_local [281.52625929]  
Right: 281.4584615384615  
Intercept 281.45104097976406  
Prediction\_local [281.06663114]  
Right: 281.03622222222214  
Intercept 281.4592200087808  
Prediction\_local [281.03143046]  
Right: 281.03622222222214  
Intercept 281.53863724016355  
Prediction\_local [281.12775792]  
Right: 281.03622222222214  
Intercept 281.5588406529695  
Prediction\_local [281.04760046]  
Right: 281.03622222222214  
Intercept 281.4966136081286  
Prediction\_local [281.05697219]  
Right: 281.03622222222214  
Intercept 281.41799546586947  
Prediction\_local [281.50902109]  
Right: 281.5777777777778  
Intercept 281.36404632961035  
Prediction\_local [281.72584995]  
Right: 281.5777777777778  
Intercept 281.29953319837193  
Prediction\_local [281.77564471]  
Right: 281.9489655172413  
Intercept 281.16443239015365  
Prediction\_local [281.77558839]  
Right: 281.9489655172413  
Intercept 281.16457486763045  
Prediction\_local [281.94929973]  
Right: 281.9489655172413  
Intercept 281.3017643379883  
Prediction\_local [281.75193185]  
Right: 281.4584615384615  
Intercept 281.2328924966934  
Prediction\_local [281.78620526]  
Right: 281.4584615384615  
Intercept 281.4956987309085  
Prediction\_local [281.11604917]  
Right: 281.03622222222214  
Intercept 281.50143037555404  
Prediction\_local [280.97547728]  
Right: 281.03622222222214  
Intercept 281.59721677810444  
Prediction\_local [281.07842195]  
Right: 281.03622222222214  
Intercept 281.57074044546  
Prediction\_local [281.01971132]  
Right: 281.03622222222214  
Intercept 281.59251808970873  
Prediction\_local [280.91147063]  
Right: 281.03622222222214  
Intercept 281.51476335752665  
Prediction\_local [281.20204497]  
Right: 281.03622222222214  
Intercept 281.47342117368584  
Prediction\_local [281.19706134]  
Right: 281.9489655172413  
Intercept 281.4233698482835  
Prediction\_local [281.64772396]  
Right: 281.5777777777778  
Intercept 281.3685112553727  
Prediction\_local [281.77447962]  
Right: 281.9489655172413  
Intercept 281.21413711148847  
Prediction\_local [281.8506335]  
Right: 281.9489655172413  
Intercept 281.27932081848974  
Prediction\_local [281.71131714]  
Right: 281.76916666666665  
Intercept 281.27082830894483  
Prediction\_local [281.73312626]  
Right: 281.76916666666665  
Intercept 281.56709378228885  
Prediction\_local [281.05501876]  
Right: 281.4584615384615  
Intercept 281.56215961709484  
Prediction\_local [280.98733112]  
Right: 281.03622222222214  
Intercept 281.4815322292097  
Prediction\_local [281.18541069]  
Right: 281.03622222222214  
Intercept 281.513304708774  
Prediction\_local [281.17723655]  
Right: 281.03622222222214  
Intercept 281.50955695982225  
Prediction\_local [281.21877441]  
Right: 281.5777777777778  
Intercept 281.5188813992734  
Prediction\_local [281.06002699]  
Right: 281.03622222222214  
Intercept 281.37356747435103  
Prediction\_local [281.69361584]  
Right: 281.5777777777778  
Intercept 281.2051911873102  
Prediction\_local [281.89718841]  
Right: 281.9489655172413  
Intercept 281.29079281991807  
Prediction\_local [281.84953796]  
Right: 281.9489655172413  
Intercept 281.34920175954113  
Prediction\_local [281.8278098]  
Right: 281.9489655172413  
Intercept 281.3682454598201  
Prediction\_local [281.66940027]  
Right: 281.76916666666665  
Intercept 281.2938001659345  
Prediction\_local [281.7298059]  
Right: 281.76916666666665  
Intercept 281.47225674619193  
Prediction\_local [281.17196478]  
Right: 281.03622222222214  
Intercept 281.53510063605285  
Prediction\_local [281.00257646]  
Right: 281.03622222222214  
Intercept 281.51992677895936  
Prediction\_local [281.09242698]  
Right: 281.03622222222214  
Intercept 281.5215730082664  
Prediction\_local [281.22975853]  
Right: 281.03622222222214  
Intercept 281.53011584442964  
Prediction\_local [281.12865898]  
Right: 281.03622222222214  
Intercept 281.52741497237304  
Prediction\_local [281.15369954]  
Right: 281.03622222222214  
Intercept 281.36627139071106  
Prediction\_local [281.64413779]  
Right: 281.5777777777778  
Intercept 281.25136304362553  
Prediction\_local [281.76770001]  
Right: 281.9489655172413  
Intercept 281.2187088411622  
Prediction\_local [281.89165098]  
Right: 281.9489655172413  
Intercept 281.21275943412655  
Prediction\_local [281.80945823]  
Right: 281.9489655172413

exp

<lime.explanation.Explanation at 0x2da67de1e50>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('AT\_G <= 25.38', 0.535989551340496),  
 ('P\_G <= 0.00', 0.10893289363142271),  
 ('9.70 < E\_G <= 13.62', -0.04822364693974687)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.535989551340496),  
 (2, -0.10893289363142271),  
 (0, 0.04822364693974687)],  
 1: [(1, 0.535989551340496),  
 (2, 0.10893289363142271),  
 (0, -0.04822364693974687)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.80945823]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.9489655172413

# Saving fe# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf2GR\_Whole.html')

##########################################################################################

#The rf best parameters are: {'max\_depth': 6, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 5, 'min\_samples\_split': 8, 'n\_estimators': 1}

########### Fitting the model on the training dataset using the best parameters  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [1],  
 'max\_depth' : [6],  
 'min\_samples\_leaf' : [5],  
 'min\_samples\_split' : [8],  
 'max\_leaf\_nodes' : [5],  
 }  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf2 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1 )  
rf2.fit(X\_G\_train, y\_R\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': [6], 'max\_leaf\_nodes': [5],  
 'min\_samples\_leaf': [5], 'min\_samples\_split': [8],  
 'n\_estimators': [1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The rf best score is:', rf2.best\_score\_)

The rf best score is: -0.11548759892362637

# Evaluation of the performance of the model on the training dataset:  
rf\_y\_predtr = rf2.predict(X\_G\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, rf\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, rf\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, rf\_y\_predtr))

The training MAE is: 1.6198052770815996  
The training MSE is: 2.7976857334908876  
The training R2 Score is: -8.175864600594384

# Evaluation of the performance of the model on the testing dataset:  
rf\_y\_predts = rf2.predict(X\_G\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, rf\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, rf\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, rf\_y\_predts))

The testing MAE is: 1.5116647869674222  
The testing MSE is: 2.490059735495422  
The testing R2 Score is: -8.080117705788055

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_rf2tr = cross\_val\_score(rf2, X\_G\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_rf2tr)))

The mean training CV score is: 0.11970635143377717

############# On the testing dataset   
score\_rf2ts = cross\_val\_score(rf2, X\_G\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_rf2ts)))

The mean testing CV score is: 0.1575433481718344

#####################LIME ##############################

###### Initializing the explainer on the training dataset  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, rf2.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.48431231668786  
Prediction\_local [281.16333219]  
Right: 281.22526315789474  
Intercept 281.44258572862014  
Prediction\_local [280.82299983]  
Right: 280.8363157894737  
Intercept 281.26280975543415  
Prediction\_local [281.84319764]  
Right: 281.929  
Intercept 281.41674661570767  
Prediction\_local [281.08522006]  
Right: 281.22526315789474  
Intercept 281.28894418119586  
Prediction\_local [281.57902449]  
Right: 281.4971428571429  
Intercept 281.3187189389898  
Prediction\_local [281.67602415]  
Right: 281.929  
Intercept 281.2018983004909  
Prediction\_local [281.50430542]  
Right: 281.4971428571429  
Intercept 281.34350517787027  
Prediction\_local [280.95926219]  
Right: 280.8363157894737  
Intercept 281.4117583836106  
Prediction\_local [280.88639102]  
Right: 280.8363157894737  
Intercept 281.47062918271905  
Prediction\_local [281.21215463]  
Right: 281.22526315789474  
Intercept 281.25604244085486  
Prediction\_local [281.5304681]  
Right: 281.4971428571429  
Intercept 281.3964912130715  
Prediction\_local [281.01197658]  
Right: 280.8363157894737  
Intercept 281.4393078198527  
Prediction\_local [281.05784436]  
Right: 281.39625  
Intercept 281.2727189097117  
Prediction\_local [281.2117741]  
Right: 281.22526315789474  
Intercept 281.23828449935473  
Prediction\_local [281.77139104]  
Right: 281.929  
Intercept 281.4806282000329  
Prediction\_local [280.8686541]  
Right: 280.8363157894737  
Intercept 281.10425588232135  
Prediction\_local [281.770304]  
Right: 281.929  
Intercept 281.31715147291305  
Prediction\_local [281.56262884]  
Right: 281.4971428571429  
Intercept 281.30423079904176  
Prediction\_local [281.39328292]  
Right: 280.8363157894737  
Intercept 281.2905341236564  
Prediction\_local [281.52190951]  
Right: 281.4971428571429  
Intercept 281.27013294305425  
Prediction\_local [281.20808669]  
Right: 281.22526315789474  
Intercept 281.16015072153607  
Prediction\_local [281.76892372]  
Right: 281.929  
Intercept 281.4187828394281  
Prediction\_local [281.1394772]  
Right: 281.22526315789474  
Intercept 281.5502083805689  
Prediction\_local [280.74238519]  
Right: 280.8363157894737  
Intercept 281.3231031355599  
Prediction\_local [281.13897897]  
Right: 280.8363157894737  
Intercept 281.2519281707594  
Prediction\_local [281.45600127]  
Right: 281.39625  
Intercept 281.4672149393431  
Prediction\_local [280.92726669]  
Right: 280.8363157894737  
Intercept 281.34912059555336  
Prediction\_local [281.49947349]  
Right: 281.39625  
Intercept 281.42070910260657  
Prediction\_local [281.19875909]  
Right: 281.22526315789474  
Intercept 281.2637808063872  
Prediction\_local [281.53542908]  
Right: 281.39625  
Intercept 281.43723368929574  
Prediction\_local [280.96332488]  
Right: 280.8363157894737  
Intercept 281.25617802711054  
Prediction\_local [281.77063639]  
Right: 281.929  
Intercept 281.2761313198592  
Prediction\_local [281.46017345]  
Right: 281.39625  
Intercept 281.20513915546496  
Prediction\_local [281.67627507]  
Right: 281.929  
Intercept 281.23240786645243  
Prediction\_local [281.7446493]  
Right: 281.929  
Intercept 281.3818893441231  
Prediction\_local [281.22830773]  
Right: 281.22526315789474  
Intercept 281.2199222334801  
Prediction\_local [281.44197228]  
Right: 281.39625  
Intercept 281.2676727817265  
Prediction\_local [281.72820329]  
Right: 281.929  
Intercept 281.4316344787441  
Prediction\_local [280.90103359]  
Right: 280.8363157894737  
Intercept 281.22412460604954  
Prediction\_local [281.59013972]  
Right: 281.4971428571429  
Intercept 281.1656201279313  
Prediction\_local [281.7635615]  
Right: 281.929  
Intercept 281.1526499484833  
Prediction\_local [281.68109485]  
Right: 281.4971428571429  
Intercept 281.2707026632836  
Prediction\_local [281.72074374]  
Right: 281.929  
Intercept 281.2030349528205  
Prediction\_local [281.80528849]  
Right: 281.929  
Intercept 281.3909914318878  
Prediction\_local [280.98218982]  
Right: 280.8363157894737  
Intercept 281.3420570455777  
Prediction\_local [281.1846045]  
Right: 281.22526315789474  
Intercept 281.1413549056064  
Prediction\_local [281.64224587]  
Right: 281.929  
Intercept 281.29748954906233  
Prediction\_local [281.49509282]  
Right: 281.39625  
Intercept 281.18966013609594  
Prediction\_local [281.44520045]  
Right: 281.39625  
Intercept 281.4119066169009  
Prediction\_local [281.20123933]  
Right: 281.4971428571429  
Intercept 281.48304049188124  
Prediction\_local [280.78474864]  
Right: 280.8363157894737  
Intercept 281.22143259573545  
Prediction\_local [281.74797716]  
Right: 281.929  
Intercept 281.3189165606005  
Prediction\_local [281.56226304]  
Right: 281.39625  
Intercept 281.3985987635881  
Prediction\_local [281.04807764]  
Right: 280.8363157894737  
Intercept 281.230150253034  
Prediction\_local [281.4748944]  
Right: 281.4971428571429  
Intercept 281.24405079282826  
Prediction\_local [281.3307233]  
Right: 281.22526315789474  
Intercept 281.3705646387578  
Prediction\_local [280.95510283]  
Right: 280.8363157894737  
Intercept 281.3060929394844  
Prediction\_local [281.02565981]  
Right: 281.22526315789474  
Intercept 281.48114652511396  
Prediction\_local [281.01560198]  
Right: 281.22526315789474  
Intercept 281.5380242394361  
Prediction\_local [280.99331121]  
Right: 280.8363157894737  
Intercept 281.38388426976246  
Prediction\_local [281.00006448]  
Right: 280.8363157894737  
Intercept 281.4933513674751  
Prediction\_local [280.99969916]  
Right: 280.8363157894737  
Intercept 281.44577398377476  
Prediction\_local [280.84626546]  
Right: 280.8363157894737  
Intercept 281.23038624436793  
Prediction\_local [281.5497645]  
Right: 281.39625  
Intercept 281.2995816491904  
Prediction\_local [281.55887183]  
Right: 281.39625  
Intercept 281.3331178505535  
Prediction\_local [281.39710756]  
Right: 281.4971428571429  
Intercept 281.254262005  
Prediction\_local [281.4900204]  
Right: 281.39625  
Intercept 281.454697474144  
Prediction\_local [280.86544643]  
Right: 280.8363157894737  
Intercept 281.4124516270724  
Prediction\_local [281.12305583]  
Right: 281.22526315789474  
Intercept 281.3499265449086  
Prediction\_local [281.09514538]  
Right: 280.8363157894737  
Intercept 281.3641136404676  
Prediction\_local [280.95396855]  
Right: 280.8363157894737  
Intercept 281.0524470383729  
Prediction\_local [281.77957289]  
Right: 281.929  
Intercept 281.3559254137033  
Prediction\_local [281.01332312]  
Right: 280.8363157894737  
Intercept 281.4878502019912  
Prediction\_local [281.39552387]  
Right: 281.22526315789474  
Intercept 281.3167079390728  
Prediction\_local [281.23270022]  
Right: 280.8363157894737  
Intercept 281.12690512612045  
Prediction\_local [281.81783968]  
Right: 281.929  
Intercept 281.28653864015587  
Prediction\_local [281.86070972]  
Right: 281.929  
Intercept 281.2599088278853  
Prediction\_local [281.7135369]  
Right: 281.4971428571429  
Intercept 281.21752968146023  
Prediction\_local [281.62114713]  
Right: 281.4971428571429  
Intercept 281.2007197548495  
Prediction\_local [281.67780435]  
Right: 281.929  
Intercept 281.443754132386  
Prediction\_local [281.49070248]  
Right: 281.39625

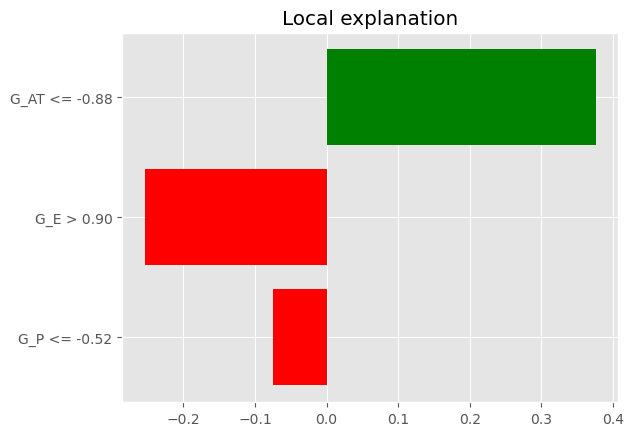
exp

<lime.explanation.Explanation at 0x25d936e8f70>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= -0.88', 0.3761513438855035),  
 ('G\_E > 0.90', -0.2536628886815333),  
 ('G\_P <= -0.52', -0.07554010685694201)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.3761513438855035),  
 (0, 0.2536628886815333),  
 (2, 0.07554010685694201)],  
 1: [(1, 0.3761513438855035),  
 (0, -0.2536628886815333),  
 (2, -0.07554010685694201)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.49070248]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.39625

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf2GR\_train.html')

#########################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, rf2.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 281.1736674079892  
Prediction\_local [281.79530031]  
Right: 281.929  
Intercept 281.26786727710197  
Prediction\_local [281.87281657]  
Right: 281.929  
Intercept 281.47347252956985  
Prediction\_local [280.79564379]  
Right: 280.8363157894737  
Intercept 281.25745716141574  
Prediction\_local [281.52135914]  
Right: 281.4971428571429  
Intercept 281.34298591981656  
Prediction\_local [281.27413964]  
Right: 281.22526315789474  
Intercept 281.447662930262  
Prediction\_local [281.22379205]  
Right: 281.22526315789474  
Intercept 281.17648589709523  
Prediction\_local [281.54322661]  
Right: 281.4971428571429  
Intercept 281.3958625856323  
Prediction\_local [281.46287294]  
Right: 281.39625  
Intercept 281.2768803560264  
Prediction\_local [281.556392]  
Right: 281.4971428571429  
Intercept 281.2530963687102  
Prediction\_local [281.40080538]  
Right: 281.39625  
Intercept 281.3817936606568  
Prediction\_local [281.03068547]  
Right: 280.8363157894737  
Intercept 281.29713816999623  
Prediction\_local [281.52357595]  
Right: 281.4971428571429  
Intercept 281.2357438558449  
Prediction\_local [281.38949109]  
Right: 281.39625  
Intercept 281.25067250361104  
Prediction\_local [281.64296584]  
Right: 281.929  
Intercept 281.14137450150866  
Prediction\_local [281.11652445]  
Right: 281.22526315789474  
Intercept 281.4330871202518  
Prediction\_local [280.8512738]  
Right: 280.8363157894737  
Intercept 281.0732253772572  
Prediction\_local [281.77855768]  
Right: 281.4971428571429  
Intercept 281.23508201967826  
Prediction\_local [281.7313646]  
Right: 281.929  
Intercept 281.47735474892465  
Prediction\_local [281.17753615]  
Right: 281.22526315789474  
Intercept 281.39479598353444  
Prediction\_local [281.07436492]  
Right: 281.22526315789474  
Intercept 281.37479599095064  
Prediction\_local [281.66071219]  
Right: 281.929  
Intercept 281.3887516810766  
Prediction\_local [281.1348102]  
Right: 281.22526315789474  
Intercept 281.31424943031686  
Prediction\_local [281.49511114]  
Right: 281.39625  
Intercept 281.2058754262874  
Prediction\_local [281.36411018]  
Right: 281.39625  
Intercept 281.2577556445146  
Prediction\_local [281.7157363]  
Right: 281.929  
Intercept 281.2870238257433  
Prediction\_local [280.86766161]  
Right: 280.8363157894737  
Intercept 281.2506870944196  
Prediction\_local [281.48083214]  
Right: 281.4971428571429

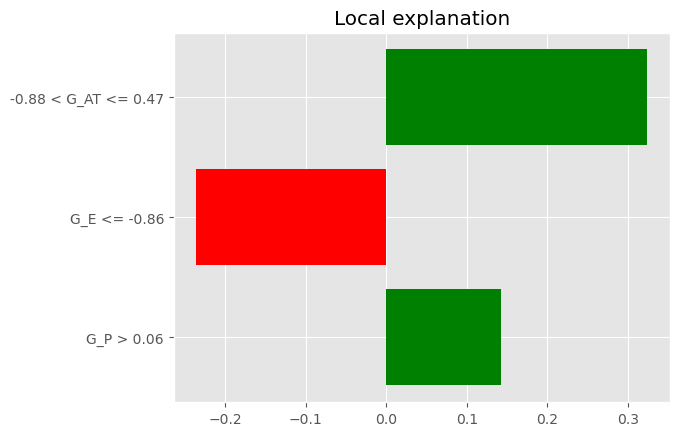
exp

<lime.explanation.Explanation at 0x25d93776fd0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('-0.88 < G\_AT <= 0.47', 0.3240793121713543),  
 ('G\_E <= -0.86', -0.23581844067586327),  
 ('G\_P > 0.06', 0.14188417576035983)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.3240793121713543),  
 (0, 0.23581844067586327),  
 (2, -0.14188417576035983)],  
 1: [(1, 0.3240793121713543),  
 (0, -0.23581844067586327),  
 (2, 0.14188417576035983)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.48083214]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.4971428571429

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf2GR\_test.html')

###################################################################################################################

###################################################################################################################

###################################################################################################################

############ MODEL 3: RS Features (X\_R) and GT Target (LL\_G)###############

########### On the whole dataset  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [1, 50, 100, 150, 200],  
 'max\_depth' : ['None', 3, 6, 9, 12, 15],  
 'min\_samples\_leaf' : [1, 2, 3, 4, 5],  
 'min\_samples\_split' : [2, 4, 6, 8, 10, 12],  
 'max\_leaf\_nodes' : ['None', 1, 2, 3, 4, 5],  
 }  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf3 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
rf3.fit(X\_R, y\_G)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': ['None', 3, 6, 9, 12, 15],  
 'max\_leaf\_nodes': ['None', 1, 2, 3, 4, 5],  
 'min\_samples\_leaf': [1, 2, 3, 4, 5],  
 'min\_samples\_split': [2, 4, 6, 8, 10, 12],  
 'n\_estimators': [1, 50, 100, 150, 200]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The rf best parameters are:', rf3.best\_params\_)  
print('The rf best score is:', rf3.best\_score\_)

The rf best parameters are: {'max\_depth': 12, 'max\_leaf\_nodes': 3, 'min\_samples\_leaf': 3, 'min\_samples\_split': 12, 'n\_estimators': 1}  
The rf best score is: -0.13978661102563764

#The rf best parameters are: {'max\_depth': 12, 'max\_leaf\_nodes': 3, 'min\_samples\_leaf': 3, 'min\_samples\_split': 12, 'n\_estimators': 1}  
#The rf best score is: -0.13978661102563764

# Evaluation of the performance of the model on the whole dataset:  
rf\_y\_pred = rf3.predict(X\_R)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_G, rf\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_G, rf\_y\_pred))  
print('R2 Score:', r2\_score(y\_G, rf\_y\_pred))

Mean Absolute Error: 0.29970886752136827  
Mean Squared Error: 0.14241087218181422  
R2 Score: 0.5315550207175568

############## LIME ##############

###### Initializing the explainer using X\_R  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the full XG datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, rf3.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 279.74824785932674  
Prediction\_local [279.68868193]  
Right: 280.24153846153837  
Intercept 279.5212151558932  
Prediction\_local [280.23734998]  
Right: 280.24153846153837  
Intercept 279.6785567507076  
Prediction\_local [279.67515114]  
Right: 279.6262499999999  
Intercept 279.7601959944698  
Prediction\_local [279.64947998]  
Right: 279.6262499999999  
Intercept 279.7176168316802  
Prediction\_local [279.59211791]  
Right: 279.6262499999999  
Intercept 279.6734610763028  
Prediction\_local [279.69050059]  
Right: 279.6262499999999  
Intercept 279.90654664936386  
Prediction\_local [279.2638213]  
Right: 279.14153846153846  
Intercept 279.7927612972469  
Prediction\_local [279.32972287]  
Right: 279.14153846153846  
Intercept 279.7165571328695  
Prediction\_local [279.57570766]  
Right: 279.6262499999999  
Intercept 279.79898151885806  
Prediction\_local [279.54510947]  
Right: 279.6262499999999  
Intercept 279.54947821074086  
Prediction\_local [280.18992837]  
Right: 280.24153846153837  
Intercept 279.5641061716882  
Prediction\_local [280.18393121]  
Right: 280.24153846153837  
Intercept 279.7010730386658  
Prediction\_local [279.78648534]  
Right: 280.24153846153837  
Intercept 279.5818103491204  
Prediction\_local [280.18128812]  
Right: 280.24153846153837  
Intercept 279.7411113049839  
Prediction\_local [279.54834549]  
Right: 279.6262499999999  
Intercept 279.76110686332044  
Prediction\_local [279.58766294]  
Right: 279.6262499999999  
Intercept 279.7234773966049  
Prediction\_local [279.67060703]  
Right: 279.6262499999999  
Intercept 279.793362538783  
Prediction\_local [279.48823103]  
Right: 279.6262499999999  
Intercept 279.7829158267464  
Prediction\_local [279.44725197]  
Right: 279.6262499999999  
Intercept 279.84528943457127  
Prediction\_local [279.30959248]  
Right: 279.14153846153846  
Intercept 279.7491889549767  
Prediction\_local [279.67475775]  
Right: 279.6262499999999  
Intercept 279.75714816934345  
Prediction\_local [279.60874522]  
Right: 279.6262499999999  
Intercept 279.5195108238301  
Prediction\_local [280.15837205]  
Right: 280.24153846153837  
Intercept 279.5431770718179  
Prediction\_local [280.21601242]  
Right: 280.24153846153837  
Intercept 279.67419194536154  
Prediction\_local [279.73566764]  
Right: 280.24153846153837  
Intercept 279.51979156606876  
Prediction\_local [280.20929706]  
Right: 280.24153846153837  
Intercept 279.6863760323024  
Prediction\_local [279.62047211]  
Right: 279.6262499999999  
Intercept 279.68721758055864  
Prediction\_local [279.52487246]  
Right: 279.6262499999999  
Intercept 279.81557910793515  
Prediction\_local [279.4993441]  
Right: 279.6262499999999  
Intercept 279.657504037595  
Prediction\_local [279.69817157]  
Right: 279.6262499999999  
Intercept 279.8373305077369  
Prediction\_local [279.42142386]  
Right: 279.6262499999999  
Intercept 279.69227170468776  
Prediction\_local [279.48785512]  
Right: 279.14153846153846  
Intercept 279.8723346468838  
Prediction\_local [279.34126049]  
Right: 279.6262499999999  
Intercept 279.69843540499147  
Prediction\_local [279.65762497]  
Right: 279.6262499999999  
Intercept 279.5259498923646  
Prediction\_local [280.17526346]  
Right: 280.24153846153837  
Intercept 279.55889982128156  
Prediction\_local [280.19196518]  
Right: 280.24153846153837  
Intercept 279.68692668913656  
Prediction\_local [279.63091604]  
Right: 280.24153846153837  
Intercept 279.5679827997162  
Prediction\_local [280.20282504]  
Right: 280.24153846153837  
Intercept 279.74104835556705  
Prediction\_local [279.60364141]  
Right: 279.6262499999999  
Intercept 279.6612600649069  
Prediction\_local [279.66288992]  
Right: 279.6262499999999  
Intercept 279.64939297390373  
Prediction\_local [279.67911347]  
Right: 279.6262499999999  
Intercept 279.8734016508726  
Prediction\_local [279.35475828]  
Right: 279.6262499999999  
Intercept 279.85057593754766  
Prediction\_local [279.26623887]  
Right: 279.14153846153846  
Intercept 279.85654438354084  
Prediction\_local [279.21902085]  
Right: 279.14153846153846  
Intercept 279.81233844228467  
Prediction\_local [279.45910284]  
Right: 279.6262499999999  
Intercept 279.63849602661816  
Prediction\_local [279.70158451]  
Right: 279.6262499999999  
Intercept 279.5355595438133  
Prediction\_local [280.14314895]  
Right: 280.24153846153837  
Intercept 279.5260761307431  
Prediction\_local [280.26814828]  
Right: 280.24153846153837  
Intercept 279.66199035938183  
Prediction\_local [279.73024979]  
Right: 280.24153846153837  
Intercept 279.5085824001709  
Prediction\_local [280.21631163]  
Right: 280.24153846153837  
Intercept 279.6669798767301  
Prediction\_local [279.80729477]  
Right: 279.6262499999999  
Intercept 279.76565537975  
Prediction\_local [279.67232909]  
Right: 279.6262499999999  
Intercept 279.75188185774994  
Prediction\_local [279.56916232]  
Right: 279.6262499999999  
Intercept 279.84165809081367  
Prediction\_local [279.2825065]  
Right: 279.6262499999999  
Intercept 279.83671222222864  
Prediction\_local [279.36565001]  
Right: 279.14153846153846  
Intercept 279.79331980792654  
Prediction\_local [279.50968165]  
Right: 279.14153846153846  
Intercept 279.8421521886252  
Prediction\_local [279.27820712]  
Right: 279.6262499999999  
Intercept 279.7300118535999  
Prediction\_local [279.67556083]  
Right: 279.6262499999999  
Intercept 279.55811656831116  
Prediction\_local [280.22345116]  
Right: 280.24153846153837  
Intercept 279.5124517788382  
Prediction\_local [280.22565032]  
Right: 280.24153846153837  
Intercept 279.75321927727487  
Prediction\_local [279.76915082]  
Right: 280.24153846153837  
Intercept 279.53115183703494  
Prediction\_local [280.20379898]  
Right: 280.24153846153837  
Intercept 279.76015768978306  
Prediction\_local [279.56438573]  
Right: 279.6262499999999  
Intercept 279.68841441529605  
Prediction\_local [279.693277]  
Right: 279.6262499999999  
Intercept 279.6740509975367  
Prediction\_local [279.66263737]  
Right: 279.6262499999999  
Intercept 279.7644981470384  
Prediction\_local [279.47413491]  
Right: 279.6262499999999  
Intercept 279.84835717820215  
Prediction\_local [279.34594646]  
Right: 279.14153846153846  
Intercept 279.82316372443614  
Prediction\_local [279.27488757]  
Right: 279.14153846153846  
Intercept 279.8574773138869  
Prediction\_local [279.35308359]  
Right: 279.6262499999999  
Intercept 279.7707711133342  
Prediction\_local [279.62827373]  
Right: 279.6262499999999  
Intercept 279.5218453460797  
Prediction\_local [280.26090598]  
Right: 280.24153846153837  
Intercept 279.5954427404828  
Prediction\_local [280.19461158]  
Right: 280.24153846153837  
Intercept 279.76348248462415  
Prediction\_local [279.69165488]  
Right: 280.24153846153837  
Intercept 279.57611605723514  
Prediction\_local [280.2479944]  
Right: 280.24153846153837  
Intercept 279.7897198186121  
Prediction\_local [279.54748462]  
Right: 279.6262499999999  
Intercept 279.79997074237826  
Prediction\_local [279.62511152]  
Right: 279.6262499999999  
Intercept 279.6871258575077  
Prediction\_local [279.64939648]  
Right: 279.6262499999999  
Intercept 279.8229772933119  
Prediction\_local [279.39270646]  
Right: 279.6262499999999  
Intercept 279.81646056093683  
Prediction\_local [279.37924637]  
Right: 279.14153846153846  
Intercept 279.70768048477976  
Prediction\_local [279.36128682]  
Right: 279.14153846153846  
Intercept 279.79461461078444  
Prediction\_local [279.35261047]  
Right: 279.6262499999999  
Intercept 279.7932693109895  
Prediction\_local [279.45499003]  
Right: 279.6262499999999  
Intercept 279.5041399219518  
Prediction\_local [280.20656533]  
Right: 280.24153846153837  
Intercept 279.5734076626023  
Prediction\_local [280.18180523]  
Right: 280.24153846153837  
Intercept 279.66125770296196  
Prediction\_local [279.74494908]  
Right: 280.24153846153837  
Intercept 279.52609987580837  
Prediction\_local [280.22386951]  
Right: 280.24153846153837  
Intercept 279.7482161619106  
Prediction\_local [279.65203531]  
Right: 279.6262499999999  
Intercept 279.6873560081274  
Prediction\_local [279.90800805]  
Right: 279.6262499999999  
Intercept 279.7634665722178  
Prediction\_local [279.58632698]  
Right: 279.6262499999999  
Intercept 279.72523535816686  
Prediction\_local [279.64823171]  
Right: 279.6262499999999  
Intercept 279.8158524383454  
Prediction\_local [279.37098984]  
Right: 279.14153846153846  
Intercept 279.8267792614655  
Prediction\_local [279.30328649]  
Right: 279.14153846153846  
Intercept 279.8993273489005  
Prediction\_local [279.28718705]  
Right: 279.6262499999999  
Intercept 279.72889292720254  
Prediction\_local [279.56379976]  
Right: 279.6262499999999  
Intercept 279.5472591459143  
Prediction\_local [280.18956826]  
Right: 280.24153846153837  
Intercept 279.52282890172177  
Prediction\_local [280.22640927]  
Right: 280.24153846153837  
Intercept 279.6932553434056  
Prediction\_local [279.76001325]  
Right: 280.24153846153837  
Intercept 279.51944333998307  
Prediction\_local [280.22795573]  
Right: 280.24153846153837  
Intercept 279.51979768507573  
Prediction\_local [280.19593172]  
Right: 280.24153846153837  
Intercept 279.7260777882361  
Prediction\_local [279.59583052]  
Right: 279.6262499999999  
Intercept 279.72153530245924  
Prediction\_local [279.69785341]  
Right: 279.6262499999999  
Intercept 279.8062550467904  
Prediction\_local [279.60143662]  
Right: 279.6262499999999  
Intercept 279.82205969576347  
Prediction\_local [279.54697326]  
Right: 279.6262499999999  
Intercept 279.78002808455193  
Prediction\_local [279.55185646]  
Right: 279.6262499999999  
Intercept 279.7978436577835  
Prediction\_local [279.66526719]  
Right: 279.6262499999999  
Intercept 279.6752698269486  
Prediction\_local [279.65447567]  
Right: 279.6262499999999  
Intercept 279.5382552181011  
Prediction\_local [280.23752503]  
Right: 280.24153846153837  
Intercept 279.5913144211475  
Prediction\_local [279.77731829]  
Right: 280.24153846153837

exp

<lime.explanation.Explanation at 0x14726efe2d0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('0.79 < E\_R <= 7.12', 0.1176587782039401),  
 ('AT\_R <= 26.86', 0.04735687163849455),  
 ('0.05 < P\_R <= 3.71', 0.02098821783989593)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(0, -0.1176587782039401),  
 (1, -0.04735687163849455),  
 (2, -0.02098821783989593)],  
 1: [(0, 0.1176587782039401),  
 (1, 0.04735687163849455),  
 (2, 0.02098821783989593)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.77731829]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.24153846153837

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf3RG\_Whole.html')

#################################################################################################

#The rf best parameters are: {'max\_depth': 12, 'max\_leaf\_nodes': 3, 'min\_samples\_leaf': 3, 'min\_samples\_split': 12, 'n\_estimators': 1}

########### Fitting the model on the training dataset using the best parameters  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [1],  
 'max\_depth' : [12],  
 'min\_samples\_leaf' : [3],  
 'min\_samples\_split' : [12],  
 'max\_leaf\_nodes' : [3],  
 }  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf3 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1 )  
rf3.fit(X\_R\_train, y\_G\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': [12], 'max\_leaf\_nodes': [3],  
 'min\_samples\_leaf': [3], 'min\_samples\_split': [12],  
 'n\_estimators': [1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The rf best score is:', rf3.best\_score\_)

The rf best score is: -0.17605144506336398

# Evaluation of the performance of the model on the training dataset:  
rf\_y\_predtr = rf3.predict(X\_R\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, rf\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, rf\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, rf\_y\_predtr))

The training MAE is: 0.29312087281079463  
The training MSE is: 0.13838755019087037  
The training R2 Score is: 0.5461157742778611

# Evaluation of the performance of the model on the testing dataset:  
rf\_y\_predts = rf3.predict(X\_R\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, rf\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, rf\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, rf\_y\_predts))

The testing MAE is: 0.33290229562220686  
The testing MSE is: 0.18490536675250369  
The testing R2 Score is: 0.3257340494240809

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_rf3tr = cross\_val\_score(rf3, X\_R\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_rf3tr)))

The mean training CV score is: 0.17991119123206417

############# On the testing dataset   
score\_rf3ts = cross\_val\_score(rf3, X\_R\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_rf3ts)))

The mean testing CV score is: 0.18659466573544634

############## LIME ##############

###### Initializing the explainer using the training dataset  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, rf3.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 279.532083364874  
Prediction\_local [279.58170037]  
Right: 280.2321739130434  
Intercept 279.75995103515777  
Prediction\_local [279.54845847]  
Right: 279.4593023255814  
Intercept 279.43250147591937  
Prediction\_local [280.13155417]  
Right: 280.2321739130434  
Intercept 279.69554166828505  
Prediction\_local [279.27075509]  
Right: 279.0200000000001  
Intercept 279.78169324200877  
Prediction\_local [279.1482041]  
Right: 279.0200000000001  
Intercept 279.40229860634355  
Prediction\_local [280.1551289]  
Right: 280.2321739130434  
Intercept 279.67456488581524  
Prediction\_local [279.18910742]  
Right: 279.0200000000001  
Intercept 279.76725215055546  
Prediction\_local [279.41549731]  
Right: 279.4593023255814  
Intercept 279.58478713084475  
Prediction\_local [279.52738708]  
Right: 279.4593023255814  
Intercept 279.8292298069482  
Prediction\_local [279.19177291]  
Right: 279.0200000000001  
Intercept 279.65355360524416  
Prediction\_local [279.43104571]  
Right: 279.4593023255814  
Intercept 279.5695232218619  
Prediction\_local [279.53501486]  
Right: 279.4593023255814  
Intercept 279.6638476353628  
Prediction\_local [279.7299484]  
Right: 280.2321739130434  
Intercept 279.75569570334534  
Prediction\_local [279.25528654]  
Right: 279.4593023255814  
Intercept 279.43641232510885  
Prediction\_local [280.11730272]  
Right: 280.2321739130434  
Intercept 279.61485360982533  
Prediction\_local [279.43569168]  
Right: 279.4593023255814  
Intercept 279.36155629023466  
Prediction\_local [280.19038374]  
Right: 280.2321739130434  
Intercept 279.70587321774104  
Prediction\_local [279.39805633]  
Right: 279.4593023255814  
Intercept 279.7215197547821  
Prediction\_local [279.35456244]  
Right: 279.4593023255814  
Intercept 279.81440340191256  
Prediction\_local [279.21978045]  
Right: 279.0200000000001  
Intercept 279.8555608198821  
Prediction\_local [279.25728931]  
Right: 279.0200000000001  
Intercept 279.4200389665248  
Prediction\_local [280.22921789]  
Right: 280.2321739130434  
Intercept 279.7041872119237  
Prediction\_local [279.50894577]  
Right: 279.4593023255814  
Intercept 279.7547810761119  
Prediction\_local [279.45260742]  
Right: 279.4593023255814  
Intercept 279.6718615363282  
Prediction\_local [279.64435529]  
Right: 280.2321739130434  
Intercept 279.67565630770844  
Prediction\_local [280.00617753]  
Right: 280.2321739130434  
Intercept 279.57902759420597  
Prediction\_local [279.40948923]  
Right: 279.4593023255814  
Intercept 279.4672825086336  
Prediction\_local [279.92088094]  
Right: 280.2321739130434  
Intercept 279.7232641163024  
Prediction\_local [279.10897603]  
Right: 279.0200000000001  
Intercept 279.3533025612697  
Prediction\_local [280.12785063]  
Right: 280.2321739130434  
Intercept 279.6893964105342  
Prediction\_local [279.33563292]  
Right: 279.4593023255814  
Intercept 279.4268207607486  
Prediction\_local [280.17070236]  
Right: 280.2321739130434  
Intercept 279.63967289618705  
Prediction\_local [279.5803753]  
Right: 280.2321739130434  
Intercept 279.4273891923862  
Prediction\_local [280.15533413]  
Right: 280.2321739130434  
Intercept 279.5098565219914  
Prediction\_local [280.13506986]  
Right: 280.2321739130434  
Intercept 279.7548765728235  
Prediction\_local [279.09149954]  
Right: 279.4593023255814  
Intercept 279.42214109032034  
Prediction\_local [280.17069395]  
Right: 280.2321739130434  
Intercept 279.6497493771893  
Prediction\_local [279.51306823]  
Right: 279.4593023255814  
Intercept 279.54498215382625  
Prediction\_local [279.806625]  
Right: 279.4593023255814  
Intercept 279.5852941471307  
Prediction\_local [279.56679255]  
Right: 279.4593023255814  
Intercept 279.4611517140746  
Prediction\_local [280.12598804]  
Right: 280.2321739130434  
Intercept 279.50795969389486  
Prediction\_local [279.64111816]  
Right: 279.4593023255814  
Intercept 279.6093858310661  
Prediction\_local [279.52481762]  
Right: 279.4593023255814  
Intercept 279.3991036557264  
Prediction\_local [280.07608296]  
Right: 280.2321739130434  
Intercept 279.7445174152943  
Prediction\_local [279.59194694]  
Right: 279.4593023255814  
Intercept 279.859686493137  
Prediction\_local [279.0464567]  
Right: 279.0200000000001  
Intercept 279.71862911313576  
Prediction\_local [279.47744215]  
Right: 279.4593023255814  
Intercept 279.5860113578336  
Prediction\_local [279.79368956]  
Right: 280.2321739130434  
Intercept 279.4410179627728  
Prediction\_local [280.14148218]  
Right: 280.2321739130434  
Intercept 279.68140784139274  
Prediction\_local [279.24915627]  
Right: 279.0200000000001  
Intercept 279.70731904798197  
Prediction\_local [279.41472373]  
Right: 279.4593023255814  
Intercept 279.44659368440256  
Prediction\_local [280.20164025]  
Right: 280.2321739130434  
Intercept 279.6701620437128  
Prediction\_local [279.72273393]  
Right: 280.2321739130434  
Intercept 279.7138876589265  
Prediction\_local [279.50739655]  
Right: 279.4593023255814  
Intercept 279.70058369103936  
Prediction\_local [279.36498325]  
Right: 279.4593023255814  
Intercept 279.57888922573716  
Prediction\_local [279.59716362]  
Right: 280.2321739130434  
Intercept 279.6485793414221  
Prediction\_local [279.64056635]  
Right: 279.4593023255814  
Intercept 279.7437380081684  
Prediction\_local [279.46683064]  
Right: 279.4593023255814  
Intercept 279.8449952435262  
Prediction\_local [279.03775503]  
Right: 279.0200000000001  
Intercept 279.7935015571745  
Prediction\_local [279.45325179]  
Right: 279.4593023255814  
Intercept 279.64403835410445  
Prediction\_local [279.34185452]  
Right: 279.4593023255814  
Intercept 279.60463416832715  
Prediction\_local [279.48119183]  
Right: 279.4593023255814  
Intercept 279.7285369638798  
Prediction\_local [279.22175642]  
Right: 279.4593023255814  
Intercept 279.3752114842828  
Prediction\_local [280.21502358]  
Right: 280.2321739130434  
Intercept 279.4578031206613  
Prediction\_local [280.09954165]  
Right: 280.2321739130434  
Intercept 279.7570638257112  
Prediction\_local [279.24473756]  
Right: 279.4593023255814  
Intercept 279.4649051437792  
Prediction\_local [280.17818851]  
Right: 280.2321739130434  
Intercept 279.8327553969626  
Prediction\_local [279.21965406]  
Right: 279.4593023255814  
Intercept 279.713299855036  
Prediction\_local [279.17791237]  
Right: 279.0200000000001  
Intercept 279.6389511035128  
Prediction\_local [279.3468573]  
Right: 279.4593023255814  
Intercept 279.6813583694209  
Prediction\_local [279.4641851]  
Right: 279.4593023255814  
Intercept 279.38364061040573  
Prediction\_local [280.22825681]  
Right: 280.2321739130434  
Intercept 279.69952993463295  
Prediction\_local [279.60095549]  
Right: 279.4593023255814  
Intercept 279.6876132202967  
Prediction\_local [279.10219538]  
Right: 279.4593023255814  
Intercept 279.62863824166396  
Prediction\_local [279.66873531]  
Right: 280.2321739130434  
Intercept 279.3790937570238  
Prediction\_local [280.15868326]  
Right: 280.2321739130434  
Intercept 279.5706625952126  
Prediction\_local [279.86733]  
Right: 280.2321739130434  
Intercept 279.5540348721493  
Prediction\_local [279.49657202]  
Right: 279.4593023255814  
Intercept 279.57417064596757  
Prediction\_local [279.58386209]  
Right: 279.4593023255814  
Intercept 279.45414200133547  
Prediction\_local [280.13434791]  
Right: 280.2321739130434  
Intercept 279.4832265294461  
Prediction\_local [280.20602801]  
Right: 280.2321739130434

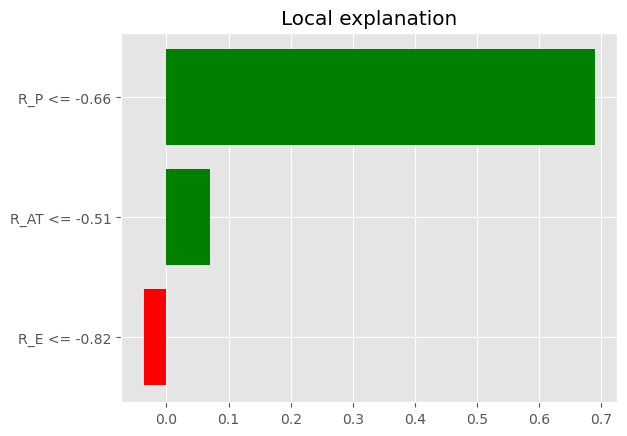
exp

<lime.explanation.Explanation at 0x25d93e22970>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P <= -0.66', 0.6895942709762946),  
 ('R\_AT <= -0.51', 0.06938385964620268),  
 ('R\_E <= -0.82', -0.036176651837259856)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.6895942709762946),  
 (1, -0.06938385964620268),  
 (0, 0.036176651837259856)],  
 1: [(2, 0.6895942709762946),  
 (1, 0.06938385964620268),  
 (0, -0.036176651837259856)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.20602801]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.2321739130434

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf3RG\_train.html')

######################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, rf3.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.3664452368447  
Prediction\_local [280.13933051]  
Right: 280.2321739130434  
Intercept 279.4965561296489  
Prediction\_local [279.90496376]  
Right: 280.2321739130434  
Intercept 279.4823364328487  
Prediction\_local [279.49295257]  
Right: 279.4593023255814  
Intercept 279.4779624511697  
Prediction\_local [279.56098016]  
Right: 279.4593023255814  
Intercept 279.3725380333551  
Prediction\_local [279.42188614]  
Right: 279.4593023255814  
Intercept 279.65850929870044  
Prediction\_local [279.57400832]  
Right: 279.4593023255814  
Intercept 279.6826466563926  
Prediction\_local [279.460444]  
Right: 279.4593023255814  
Intercept 279.55401836209614  
Prediction\_local [279.94547482]  
Right: 280.2321739130434  
Intercept 279.73365456298154  
Prediction\_local [279.14287854]  
Right: 279.0200000000001  
Intercept 279.67582766836676  
Prediction\_local [279.70139232]  
Right: 280.2321739130434  
Intercept 279.58317787775053  
Prediction\_local [279.6016324]  
Right: 279.4593023255814  
Intercept 279.4989037035967  
Prediction\_local [279.62680825]  
Right: 279.4593023255814  
Intercept 279.6053477390833  
Prediction\_local [280.14638355]  
Right: 280.2321739130434  
Intercept 279.4057459314451  
Prediction\_local [280.23565352]  
Right: 280.2321739130434  
Intercept 279.4946522208467  
Prediction\_local [279.63362313]  
Right: 280.2321739130434  
Intercept 279.5528915547146  
Prediction\_local [279.34456456]  
Right: 279.4593023255814  
Intercept 279.6364270181078  
Prediction\_local [279.43415699]  
Right: 279.4593023255814  
Intercept 279.6186302136135  
Prediction\_local [279.53769778]  
Right: 279.4593023255814  
Intercept 279.68522657096446  
Prediction\_local [279.16674824]  
Right: 279.0200000000001  
Intercept 279.6796856757967  
Prediction\_local [279.11936603]  
Right: 279.0200000000001  
Intercept 279.3641056561094  
Prediction\_local [280.18459273]  
Right: 280.2321739130434  
Intercept 279.6650541452445  
Prediction\_local [279.11033277]  
Right: 279.0200000000001  
Intercept 279.78964200272003  
Prediction\_local [279.98792222]  
Right: 280.2321739130434  
Intercept 279.46202743371373  
Prediction\_local [280.08580927]  
Right: 280.2321739130434  
Intercept 279.5038679696722  
Prediction\_local [279.99338866]  
Right: 280.2321739130434  
Intercept 279.5711283330823  
Prediction\_local [279.62256579]  
Right: 279.4593023255814  
Intercept 279.63516268408813  
Prediction\_local [279.35071061]  
Right: 279.0200000000001

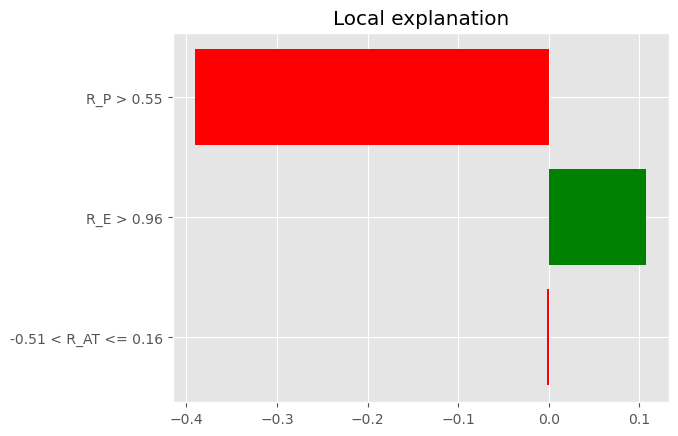
exp

<lime.explanation.Explanation at 0x25d93eb5670>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P > 0.55', -0.39003630233173125),  
 ('R\_E > 0.96', 0.1075797237381211),  
 ('-0.51 < R\_AT <= 0.16', -0.0019954969685365637)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 0.39003630233173125),  
 (0, -0.1075797237381211),  
 (1, 0.0019954969685365637)],  
 1: [(2, -0.39003630233173125),  
 (0, 0.1075797237381211),  
 (1, -0.0019954969685365637)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.35071061]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.0200000000001

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf3RG\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

############ MODEL 4: GT Features (X\_G) and GT Target (LL\_G)###############

########### On the whole dataset  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [1, 50, 100, 150, 200],  
 'max\_depth' : ['None', 3, 6, 9, 12, 15],  
 'min\_samples\_leaf' : [1, 2, 3, 4, 5],  
 'min\_samples\_split' : [2, 4, 6, 8, 10, 12],  
 'max\_leaf\_nodes' : ['None', 1, 2, 3, 4, 5],  
 }  
  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf4 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
rf4.fit(X\_G, y\_G)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': ['None', 3, 6, 9, 12, 15],  
 'max\_leaf\_nodes': ['None', 1, 2, 3, 4, 5],  
 'min\_samples\_leaf': [1, 2, 3, 4, 5],  
 'min\_samples\_split': [2, 4, 6, 8, 10, 12],  
 'n\_estimators': [1, 50, 100, 150, 200]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The rf best parameters are:', rf4.best\_params\_)  
print('The rf best score is:', rf4.best\_score\_)

The rf best parameters are: {'max\_depth': 15, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 1}  
The rf best score is: -0.09276096313074417

#The rf best parameters are: {'max\_depth': 15, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 1}  
#The rf best score is: -0.09276096313074417

# Evaluation of the performance of the model on the whole dataset:  
rf\_y\_pred = rf4.predict(X\_G)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_G, rf\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_G, rf\_y\_pred))  
print('R2 Score:', r2\_score(y\_G, rf\_y\_pred))

Mean Absolute Error: 0.230925120772938  
Mean Squared Error: 0.0975659123064785  
R2 Score: 0.6790676086111536

############### LIME #################

###### Initializing the explainer using the whole X\_G  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the whole XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, rf4.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 279.69565363649696  
Prediction\_local [280.34864627]  
Right: 280.3377142857142  
Intercept 279.5870838339844  
Prediction\_local [280.37408683]  
Right: 280.3377142857142  
Intercept 279.7585907103897  
Prediction\_local [279.5929505]  
Right: 279.99  
Intercept 279.9619552969435  
Prediction\_local [279.58588578]  
Right: 279.48879999999997  
Intercept 279.72325148862376  
Prediction\_local [279.64563712]  
Right: 279.48879999999997  
Intercept 280.0531397737274  
Prediction\_local [279.35387642]  
Right: 279.48879999999997  
Intercept 280.03217588185305  
Prediction\_local [279.36854375]  
Right: 279.1691304347825  
Intercept 279.95658238458134  
Prediction\_local [279.42019615]  
Right: 279.1691304347825  
Intercept 279.8985356388264  
Prediction\_local [279.86378198]  
Right: 279.4742857142857  
Intercept 279.6750004481021  
Prediction\_local [280.23279625]  
Right: 279.99  
Intercept 279.6587246857576  
Prediction\_local [280.30547994]  
Right: 280.3377142857142  
Intercept 279.55907147414797  
Prediction\_local [280.41952912]  
Right: 280.3377142857142  
Intercept 279.6670226397763  
Prediction\_local [280.3741932]  
Right: 280.3377142857142  
Intercept 279.54584951217424  
Prediction\_local [280.31861389]  
Right: 280.3377142857142  
Intercept 279.7505319173701  
Prediction\_local [279.871999]  
Right: 279.99  
Intercept 279.8581323937044  
Prediction\_local [279.60313148]  
Right: 279.48879999999997  
Intercept 279.84873678543823  
Prediction\_local [279.58742587]  
Right: 279.48879999999997  
Intercept 280.0254450605921  
Prediction\_local [279.5051696]  
Right: 279.48879999999997  
Intercept 280.14836469615057  
Prediction\_local [279.27039621]  
Right: 279.1691304347825  
Intercept 280.1020486578106  
Prediction\_local [279.42126031]  
Right: 279.1691304347825  
Intercept 280.024210106286  
Prediction\_local [279.74431151]  
Right: 279.4742857142857  
Intercept 279.70694495254594  
Prediction\_local [280.17904822]  
Right: 279.99  
Intercept 279.6801973748595  
Prediction\_local [280.20526189]  
Right: 280.3377142857142  
Intercept 279.4831487902518  
Prediction\_local [280.42825798]  
Right: 280.3377142857142  
Intercept 279.57380497392563  
Prediction\_local [280.38376864]  
Right: 280.3377142857142  
Intercept 279.5902242622093  
Prediction\_local [280.34436441]  
Right: 280.3377142857142  
Intercept 279.94715988788283  
Prediction\_local [279.626778]  
Right: 279.99  
Intercept 279.98031066037504  
Prediction\_local [279.50235504]  
Right: 279.48879999999997  
Intercept 279.88072219517903  
Prediction\_local [279.51634513]  
Right: 279.48879999999997  
Intercept 280.0851190101267  
Prediction\_local [279.25339283]  
Right: 279.1691304347825  
Intercept 279.95987846550514  
Prediction\_local [279.93526662]  
Right: 279.1691304347825  
Intercept 279.94492017727896  
Prediction\_local [279.88622585]  
Right: 279.1691304347825  
Intercept 279.90505584864616  
Prediction\_local [279.86633099]  
Right: 279.1691304347825  
Intercept 279.59113729788174  
Prediction\_local [280.27020239]  
Right: 279.99  
Intercept 279.5067318968638  
Prediction\_local [280.29054945]  
Right: 280.3377142857142  
Intercept 279.66201999178026  
Prediction\_local [280.40143392]  
Right: 280.3377142857142  
Intercept 279.605385866432  
Prediction\_local [280.35333028]  
Right: 280.3377142857142  
Intercept 279.66513816487566  
Prediction\_local [280.37477752]  
Right: 280.3377142857142  
Intercept 279.6882943692006  
Prediction\_local [279.86125742]  
Right: 279.99  
Intercept 279.94456721135344  
Prediction\_local [279.59487363]  
Right: 279.48879999999997  
Intercept 280.08583325712874  
Prediction\_local [279.42101521]  
Right: 279.48879999999997  
Intercept 279.9985382092079  
Prediction\_local [279.43828329]  
Right: 279.1691304347825  
Intercept 280.05658194005  
Prediction\_local [279.39020125]  
Right: 279.1691304347825  
Intercept 279.90503317042493  
Prediction\_local [279.72452904]  
Right: 279.1691304347825  
Intercept 279.85825659809586  
Prediction\_local [279.93777317]  
Right: 279.4742857142857  
Intercept 279.70212772715513  
Prediction\_local [279.90057114]  
Right: 279.99  
Intercept 279.7054692242211  
Prediction\_local [280.17239806]  
Right: 280.3377142857142  
Intercept 279.64527969074646  
Prediction\_local [280.43627364]  
Right: 280.3377142857142  
Intercept 279.55547085861855  
Prediction\_local [280.35037126]  
Right: 280.3377142857142  
Intercept 279.53780407103125  
Prediction\_local [280.42996233]  
Right: 280.3377142857142  
Intercept 279.66517783415526  
Prediction\_local [280.11652193]  
Right: 279.99  
Intercept 279.8293277822069  
Prediction\_local [279.57127025]  
Right: 279.48879999999997  
Intercept 279.9825367553637  
Prediction\_local [279.55589933]  
Right: 279.48879999999997  
Intercept 280.059029286206  
Prediction\_local [279.31052328]  
Right: 279.1691304347825  
Intercept 280.08857057066274  
Prediction\_local [279.23993688]  
Right: 279.1691304347825  
Intercept 279.9955141061161  
Prediction\_local [279.42949175]  
Right: 279.1691304347825  
Intercept 279.9380148158236  
Prediction\_local [279.7954969]  
Right: 279.4742857142857  
Intercept 279.64524439679644  
Prediction\_local [279.86678751]  
Right: 279.99  
Intercept 279.56637233386755  
Prediction\_local [280.21586171]  
Right: 280.3377142857142  
Intercept 279.48052178087204  
Prediction\_local [280.39773224]  
Right: 280.3377142857142  
Intercept 279.6211869337218  
Prediction\_local [280.45528439]  
Right: 280.3377142857142  
Intercept 279.6210829098978  
Prediction\_local [280.31002768]  
Right: 280.3377142857142  
Intercept 279.7134067268713  
Prediction\_local [279.86041781]  
Right: 279.99  
Intercept 279.98793549019246  
Prediction\_local [279.43313663]  
Right: 279.48879999999997  
Intercept 279.87677408054725  
Prediction\_local [279.55168473]  
Right: 279.48879999999997  
Intercept 280.10009556184843  
Prediction\_local [279.26540942]  
Right: 279.1691304347825  
Intercept 280.0007709028459  
Prediction\_local [279.555257]  
Right: 279.1691304347825  
Intercept 280.05042446300257  
Prediction\_local [279.69631252]  
Right: 279.1691304347825  
Intercept 279.9130205411134  
Prediction\_local [279.90060607]  
Right: 279.1691304347825  
Intercept 279.92569380185506  
Prediction\_local [280.01382219]  
Right: 280.3377142857142  
Intercept 279.66216517546866  
Prediction\_local [280.25164052]  
Right: 280.3377142857142  
Intercept 279.60161486346533  
Prediction\_local [280.36044527]  
Right: 280.3377142857142  
Intercept 279.6277790360214  
Prediction\_local [280.34756567]  
Right: 280.3377142857142  
Intercept 279.68164346292684  
Prediction\_local [280.29224408]  
Right: 280.3377142857142  
Intercept 279.56719713044885  
Prediction\_local [279.82572719]  
Right: 279.99  
Intercept 279.9388397350021  
Prediction\_local [279.56842333]  
Right: 279.48879999999997  
Intercept 279.8776750952848  
Prediction\_local [279.58360139]  
Right: 279.48879999999997  
Intercept 280.1210702756499  
Prediction\_local [279.29967086]  
Right: 279.1691304347825  
Intercept 280.0627998562783  
Prediction\_local [279.52190712]  
Right: 279.1691304347825  
Intercept 280.0648422931493  
Prediction\_local [279.29495967]  
Right: 279.1691304347825  
Intercept 280.02073407241215  
Prediction\_local [279.69229277]  
Right: 279.1691304347825  
Intercept 279.8289840156095  
Prediction\_local [280.09954005]  
Right: 280.3377142857142  
Intercept 279.720276185941  
Prediction\_local [280.25235587]  
Right: 280.3377142857142  
Intercept 279.56191783901374  
Prediction\_local [280.32321852]  
Right: 280.3377142857142  
Intercept 279.6183275844069  
Prediction\_local [280.43025025]  
Right: 280.3377142857142  
Intercept 279.5521853970317  
Prediction\_local [280.40939764]  
Right: 280.3377142857142  
Intercept 279.904849551971  
Prediction\_local [279.93634892]  
Right: 279.99  
Intercept 279.8516987910125  
Prediction\_local [279.59513943]  
Right: 279.48879999999997  
Intercept 279.8469834009991  
Prediction\_local [279.51737721]  
Right: 279.48879999999997  
Intercept 280.1047107805935  
Prediction\_local [279.1648295]  
Right: 279.4742857142857  
Intercept 280.1254959531167  
Prediction\_local [279.34449921]  
Right: 279.1691304347825  
Intercept 280.1037896095261  
Prediction\_local [279.31947187]  
Right: 279.1691304347825  
Intercept 279.9101186793779  
Prediction\_local [280.16340129]  
Right: 280.3377142857142  
Intercept 279.6045774005918  
Prediction\_local [280.1990507]  
Right: 279.99  
Intercept 279.62397005835675  
Prediction\_local [280.22181206]  
Right: 280.3377142857142  
Intercept 279.51960357494266  
Prediction\_local [280.32292074]  
Right: 280.3377142857142  
Intercept 279.5179796275415  
Prediction\_local [280.44261738]  
Right: 280.3377142857142  
Intercept 279.6237417473232  
Prediction\_local [280.39898219]  
Right: 280.3377142857142  
Intercept 279.7889046625362  
Prediction\_local [279.89451687]  
Right: 279.99  
Intercept 279.7304068716837  
Prediction\_local [279.60163838]  
Right: 279.48879999999997  
Intercept 279.92213646251594  
Prediction\_local [279.56020209]  
Right: 279.48879999999997  
Intercept 280.1191047609221  
Prediction\_local [279.67010274]  
Right: 279.4742857142857  
Intercept 280.0379205034266  
Prediction\_local [279.56299445]  
Right: 279.1691304347825  
Intercept 280.0919289939604  
Prediction\_local [279.45971848]  
Right: 279.1691304347825  
Intercept 279.92500439234215  
Prediction\_local [279.69011222]  
Right: 279.1691304347825  
Intercept 279.65914869374666  
Prediction\_local [280.29982644]  
Right: 279.99  
Intercept 279.62411459681744  
Prediction\_local [280.25887051]  
Right: 280.3377142857142  
Intercept 279.6399098230058  
Prediction\_local [280.3755209]  
Right: 280.3377142857142

exp

<lime.explanation.Explanation at 0x2da695c3250>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()

## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('AT\_G <= 25.38', 0.5164132092762233),  
 ('P\_G <= 0.00', 0.20456403413753219),  
 ('9.70 < E\_G <= 13.62', 0.014633829535961585)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.5164132092762233),  
 (2, -0.20456403413753219),  
 (0, -0.014633829535961585)],  
 1: [(1, 0.5164132092762233),  
 (2, 0.20456403413753219),  
 (0, 0.014633829535961585)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.3755209]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.3377142857142

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf4GG\_Whole.html')

##############################################################################

#The rf best parameters are: {'max\_depth': 15, 'max\_leaf\_nodes': 5, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 1}

########### Fitting the model on the training dataset using the best parameters  
model = RandomForestRegressor()  
rf\_param\_grid = {'n\_estimators' : [1],  
 'max\_depth' : [15],  
 'min\_samples\_leaf' : [1],  
 'min\_samples\_split' : [2],  
 'max\_leaf\_nodes' : [5],  
 }  
rf\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
rf4 = GridSearchCV(model, rf\_param\_grid, cv = rf\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1 )  
rf4.fit(X\_G\_train, y\_G\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=RandomForestRegressor(), n\_jobs=-1,  
 param\_grid={'max\_depth': [15], 'max\_leaf\_nodes': [5],  
 'min\_samples\_leaf': [1], 'min\_samples\_split': [2],  
 'n\_estimators': [1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The rf best score is:', rf4.best\_score\_)

The rf best score is: -0.14886116353716158

# Evaluation of the performance of the model on the training dataset:  
rf\_y\_predtr = rf4.predict(X\_G\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, rf\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, rf\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, rf\_y\_predtr))

The training MAE is: 0.2171306368528611  
The training MSE is: 0.095683305274187  
The training R2 Score is: 0.6861773846779571

# Evaluation of the performance of the model on the testing dataset:  
rf\_y\_predts = rf4.predict(X\_G\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, rf\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, rf\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, rf\_y\_predts))

The testing MAE is: 0.36122327886818717  
The testing MSE is: 0.19548556781120327  
The testing R2 Score is: 0.28715285814001645

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_rf4tr = cross\_val\_score(rf4, X\_G\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_rf4tr)))

The mean training CV score is: 0.15190545221999857

############# On the testing dataset   
score\_rf4ts = cross\_val\_score(rf4, X\_G\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_rf4ts)))

The mean testing CV score is: 0.14844152374738379

############### LIME #################

###### Initializing the explainer using the training dataset  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, rf4.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 279.3968459406125  
Prediction\_local [279.79523929]  
Right: 279.90739130434775  
Intercept 279.417629720522  
Prediction\_local [279.47163985]  
Right: 279.4376923076924  
Intercept 279.3104716535748  
Prediction\_local [280.16567427]  
Right: 280.40125  
Intercept 279.83341031543756  
Prediction\_local [279.08987857]  
Right: 279.05333333333334  
Intercept 279.8856559963336  
Prediction\_local [279.20851533]  
Right: 279.05333333333334  
Intercept 279.2255290779473  
Prediction\_local [280.19726458]  
Right: 280.40125  
Intercept 279.86367245393166  
Prediction\_local [279.1995394]  
Right: 279.05333333333334  
Intercept 279.40010252561945  
Prediction\_local [279.57306159]  
Right: 279.4376923076924  
Intercept 279.43321715319615  
Prediction\_local [279.56717798]  
Right: 279.4376923076924  
Intercept 279.9306694226768  
Prediction\_local [279.07334269]  
Right: 279.05333333333334  
Intercept 279.64117279409817  
Prediction\_local [279.87750957]  
Right: 279.90739130434775  
Intercept 279.44620055711675  
Prediction\_local [279.60987851]  
Right: 279.4376923076924  
Intercept 279.47393111068374  
Prediction\_local [279.79899292]  
Right: 279.90739130434775  
Intercept 279.85672810517934  
Prediction\_local [279.10681002]  
Right: 279.05333333333334  
Intercept 279.1485889423411  
Prediction\_local [280.13333589]  
Right: 280.40125  
Intercept 279.9221155658579  
Prediction\_local [278.90834039]  
Right: 279.05333333333334  
Intercept 279.2675869660218  
Prediction\_local [280.09854641]  
Right: 279.9636363636364  
Intercept 279.83337698051514  
Prediction\_local [279.26650689]  
Right: 279.90739130434775  
Intercept 279.8611316536244  
Prediction\_local [279.25773751]  
Right: 279.4376923076924  
Intercept 279.8839649706781  
Prediction\_local [279.18039943]  
Right: 279.05333333333334  
Intercept 279.8904059416527  
Prediction\_local [279.13976344]  
Right: 279.05333333333334  
Intercept 279.29783946119113  
Prediction\_local [280.23800095]  
Right: 280.40125  
Intercept 279.46160959937305  
Prediction\_local [279.79201774]  
Right: 279.90739130434775  
Intercept 279.633813059783  
Prediction\_local [279.39978075]  
Right: 279.4376923076924  
Intercept 279.3135073486164  
Prediction\_local [279.73560533]  
Right: 279.4376923076924  
Intercept 279.3586751263936  
Prediction\_local [280.04002098]  
Right: 279.9636363636364  
Intercept 279.8758200616115  
Prediction\_local [279.46915963]  
Right: 279.4376923076924  
Intercept 279.20185527758  
Prediction\_local [280.10849862]  
Right: 279.9636363636364  
Intercept 279.85094941977536  
Prediction\_local [279.10154614]  
Right: 279.05333333333334  
Intercept 279.39191002053457  
Prediction\_local [280.04098683]  
Right: 279.9636363636364  
Intercept 279.87824862930967  
Prediction\_local [279.56019875]  
Right: 279.4376923076924  
Intercept 279.34010163015574  
Prediction\_local [280.20546666]  
Right: 280.40125  
Intercept 279.24119026635066  
Prediction\_local [280.13830331]  
Right: 279.9636363636364  
Intercept 279.22113612912045  
Prediction\_local [280.22519594]  
Right: 280.40125  
Intercept 279.33517643943276  
Prediction\_local [280.45866076]  
Right: 280.40125  
Intercept 279.6810005543161  
Prediction\_local [279.79047432]  
Right: 279.90739130434775  
Intercept 279.2260775613023  
Prediction\_local [280.00676333]  
Right: 279.9636363636364  
Intercept 279.2837251740233  
Prediction\_local [279.82474783]  
Right: 279.90739130434775  
Intercept 279.33894810435356  
Prediction\_local [279.58427699]  
Right: 279.4376923076924  
Intercept 279.3729216663044  
Prediction\_local [280.33577649]  
Right: 280.40125  
Intercept 279.2706984880596  
Prediction\_local [280.17206466]  
Right: 280.40125  
Intercept 279.7070298636187  
Prediction\_local [280.00225015]  
Right: 279.90739130434775  
Intercept 279.2754349670771  
Prediction\_local [279.90110587]  
Right: 279.90739130434775  
Intercept 279.34978424982376  
Prediction\_local [280.12499605]  
Right: 280.40125  
Intercept 279.59503097864297  
Prediction\_local [279.40149042]  
Right: 279.4376923076924  
Intercept 279.8508866915074  
Prediction\_local [279.0793026]  
Right: 279.05333333333334  
Intercept 279.2221026189848  
Prediction\_local [280.11079016]  
Right: 279.90739130434775  
Intercept 279.3248690085596  
Prediction\_local [279.94396971]  
Right: 279.90739130434775  
Intercept 279.3576750638832  
Prediction\_local [280.01785844]  
Right: 279.9636363636364  
Intercept 279.8569891312823  
Prediction\_local [279.10882777]  
Right: 279.05333333333334  
Intercept 279.6062477205346  
Prediction\_local [279.4143992]  
Right: 279.4376923076924  
Intercept 279.22880005825857  
Prediction\_local [280.11633596]  
Right: 280.40125  
Intercept 279.11499452107716  
Prediction\_local [280.10199273]  
Right: 279.9636363636364  
Intercept 279.39051655858395  
Prediction\_local [279.6603575]  
Right: 279.4376923076924  
Intercept 279.7833122825172  
Prediction\_local [279.25823659]  
Right: 279.05333333333334  
Intercept 279.41918601680567  
Prediction\_local [279.86449686]  
Right: 279.90739130434775  
Intercept 279.47719180399855  
Prediction\_local [279.5323254]  
Right: 279.4376923076924  
Intercept 279.3725389511561  
Prediction\_local [279.73698274]  
Right: 279.90739130434775  
Intercept 280.0082188308584  
Prediction\_local [279.29067444]  
Right: 279.05333333333334  
Intercept 279.4739585639135  
Prediction\_local [279.46968615]  
Right: 279.4376923076924  
Intercept 279.3978735845339  
Prediction\_local [279.53675397]  
Right: 279.4376923076924  
Intercept 279.8674364746148  
Prediction\_local [279.50651884]  
Right: 279.4376923076924  
Intercept 280.0139544650866  
Prediction\_local [278.85693929]  
Right: 279.05333333333334  
Intercept 279.27564907261814  
Prediction\_local [280.01877212]  
Right: 279.9636363636364  
Intercept 279.3049485629488  
Prediction\_local [279.88838321]  
Right: 279.9636363636364  
Intercept 279.82418579771365  
Prediction\_local [279.25769715]  
Right: 279.05333333333334  
Intercept 279.23935074733447  
Prediction\_local [279.99128615]  
Right: 279.9636363636364  
Intercept 279.9831737503129  
Prediction\_local [278.79268049]  
Right: 279.05333333333334  
Intercept 279.8839200682534  
Prediction\_local [279.12834675]  
Right: 279.05333333333334  
Intercept 279.8704729860329  
Prediction\_local [279.07113371]  
Right: 279.05333333333334  
Intercept 279.45533120635724  
Prediction\_local [279.60337479]  
Right: 279.4376923076924  
Intercept 279.2653970125246  
Prediction\_local [280.04871077]  
Right: 280.40125  
Intercept 279.4488011175676  
Prediction\_local [279.56184354]  
Right: 279.4376923076924  
Intercept 279.76316651515083  
Prediction\_local [279.95991873]  
Right: 279.90739130434775  
Intercept 279.26041352971885  
Prediction\_local [279.66143298]  
Right: 279.4376923076924  
Intercept 279.35155350820673  
Prediction\_local [280.24164109]  
Right: 280.40125  
Intercept 279.39858082820035  
Prediction\_local [280.18278018]  
Right: 279.9636363636364  
Intercept 279.24201399155015  
Prediction\_local [279.83805982]  
Right: 279.90739130434775  
Intercept 279.31438433723747  
Prediction\_local [280.08992247]  
Right: 279.90739130434775  
Intercept 279.1166803115914  
Prediction\_local [280.32021447]  
Right: 280.40125  
Intercept 279.2392070880104  
Prediction\_local [279.92593891]  
Right: 279.9636363636364

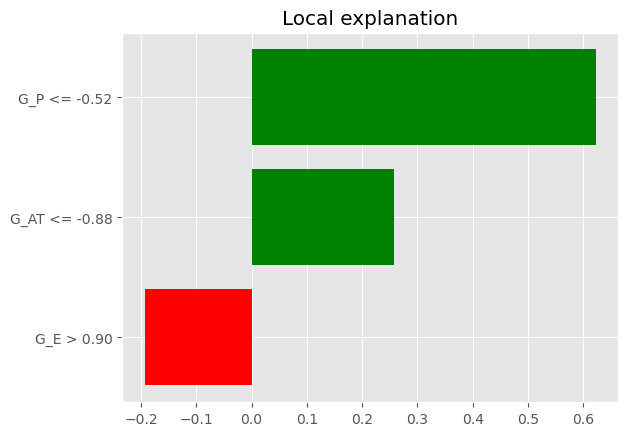
exp

<lime.explanation.Explanation at 0x25d9449ab50>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_P <= -0.52', 0.6228507743918488),  
 ('G\_AT <= -0.88', 0.2568152453294894),  
 ('G\_E > 0.90', -0.19293420199604272)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.6228507743918488),  
 (1, -0.2568152453294894),  
 (0, 0.19293420199604272)],  
 1: [(2, 0.6228507743918488),  
 (1, 0.2568152453294894),  
 (0, -0.19293420199604272)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.92593891]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.9636363636364

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf4GG\_train.html')

################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, rf4.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.0924852855373  
Prediction\_local [280.04862046]  
Right: 279.9636363636364  
Intercept 279.26324127314047  
Prediction\_local [280.14634124]  
Right: 280.40125  
Intercept 279.8345303259164  
Prediction\_local [279.40021424]  
Right: 279.4376923076924  
Intercept 279.6049270346059  
Prediction\_local [279.79496235]  
Right: 279.90739130434775  
Intercept 279.7600875640199  
Prediction\_local [279.74897532]  
Right: 279.90739130434775  
Intercept 279.685200491178  
Prediction\_local [279.85566512]  
Right: 279.90739130434775  
Intercept 279.757307397258  
Prediction\_local [279.12353156]  
Right: 279.05333333333334  
Intercept 279.7067697805803  
Prediction\_local [279.94420631]  
Right: 279.9636363636364  
Intercept 280.0520360945147  
Prediction\_local [279.3238323]  
Right: 279.05333333333334  
Intercept 279.16216381620376  
Prediction\_local [279.98026113]  
Right: 279.9636363636364  
Intercept 279.74325881467917  
Prediction\_local [279.46120556]  
Right: 279.4376923076924  
Intercept 279.3470011915149  
Prediction\_local [279.79120859]  
Right: 279.90739130434775  
Intercept 279.32918860456067  
Prediction\_local [280.02241768]  
Right: 279.9636363636364  
Intercept 279.6539078882995  
Prediction\_local [280.03695389]  
Right: 280.40125  
Intercept 279.36781715886667  
Prediction\_local [279.83148843]  
Right: 279.90739130434775  
Intercept 279.6898587212512  
Prediction\_local [279.69332976]  
Right: 279.4376923076924  
Intercept 279.8580344290307  
Prediction\_local [279.24638914]  
Right: 279.05333333333334  
Intercept 279.4866628766371  
Prediction\_local [280.11475669]  
Right: 279.90739130434775  
Intercept 279.9545852991058  
Prediction\_local [279.23064048]  
Right: 279.05333333333334  
Intercept 279.88916181694026  
Prediction\_local [279.16436452]  
Right: 279.05333333333334  
Intercept 279.2133692425102  
Prediction\_local [279.96814472]  
Right: 280.40125  
Intercept 279.84483868525336  
Prediction\_local [279.06988162]  
Right: 279.05333333333334  
Intercept 279.2951957771649  
Prediction\_local [279.86795893]  
Right: 279.9636363636364  
Intercept 279.3139672162229  
Prediction\_local [280.07752541]  
Right: 279.9636363636364  
Intercept 279.11076476287366  
Prediction\_local [280.21662823]  
Right: 280.40125  
Intercept 279.7064798315709  
Prediction\_local [279.47122825]  
Right: 279.4376923076924  
Intercept 279.74080494393843  
Prediction\_local [279.25617629]  
Right: 279.05333333333334

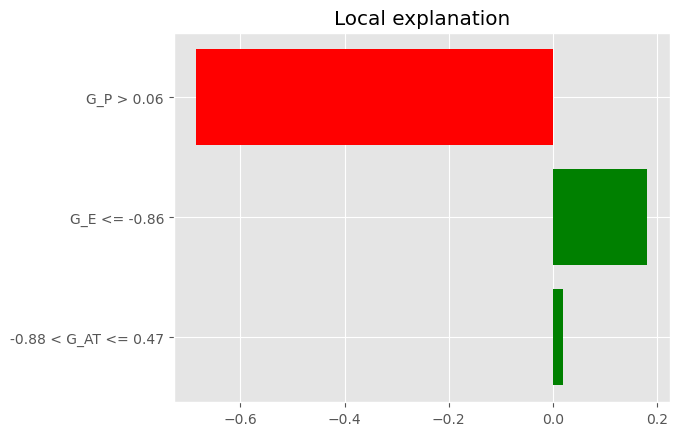
exp

<lime.explanation.Explanation at 0x25d94426280>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_P > 0.06', -0.6845549236026988),  
 ('G\_E <= -0.86', 0.18135127065689297),  
 ('-0.88 < G\_AT <= 0.47', 0.018574997703495397)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 0.6845549236026988),  
 (0, -0.18135127065689297),  
 (1, -0.018574997703495397)],  
 1: [(2, -0.6845549236026988),  
 (0, 0.18135127065689297),  
 (1, 0.018574997703495397)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.25617629]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.05333333333334

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_rf4GG\_test.html')

##################### END ####################################

#######################################################################################################

#######################################################################################################

#######################################################################################################

##################### EXTREME GRADIENT BOOSTING REGRESSION ###################

# conda install conda-forge::xgboost  
from xgboost import XGBRegressor

############ MODEL 1: RS Features (X\_R) and RS Target (LL\_R)###############

########### On the whole dataset  
model = XGBRegressor()  
xgb\_param\_grid = { 'learning\_rate' : [0.001, 0.1, 0.20, 0.30],  
 'n\_estimators' : [100, 500, 1000],  
 'subsample' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'colsample\_bytree' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'max\_depth' : [3, 6, 9, 12, 15],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb1 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb1.fit(X\_R, y\_R)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'learning\_rate': [0.001, 0.1, 0.2, 0.3],  
 'max\_depth': [3, 6, 9, 12, 15],  
 'n\_estimators': [100, 500, 1000],  
 'subsample': [0.1, 0.3, 0.5, 0.7, 0.9, 1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The best parameters are:', xgb1.best\_params\_)  
print('The best score is:', xgb1.best\_score\_)

The best parameters are: {'colsample\_bytree': 0.7, 'learning\_rate': 0.1, 'max\_depth': 9, 'n\_estimators': 100, 'subsample': 0.5}  
The best score is: -0.12347843485644353

#The best parameters are: {'colsample\_bytree': 0.7, 'learning\_rate': 0.1, 'max\_depth': 9, 'n\_estimators': 100, 'subsample': 0.5}  
#The best score is: -0.12347843485644353

# Evaluation of the performance of the model on the whole dataset:  
xgb\_y\_pred = xgb1.predict(X\_R)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_R, xgb\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_R, xgb\_y\_pred))  
print('R2 Score:', r2\_score(y\_R, xgb\_y\_pred))

Mean Absolute Error: 0.024459194607202944  
Mean Squared Error: 0.0012654563122738594  
R2 Score: 0.9937991938323314

################ LIME ################

###### Initializing the explainer on the whole features dataset  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the full datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, xgb1.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 281.3906055773314  
Prediction\_local [281.48732624]  
Right: 281.86118  
Intercept 281.3275610293468  
Prediction\_local [281.69214483]  
Right: 281.60107  
Intercept 281.39032311403224  
Prediction\_local [281.24523413]  
Right: 281.33136  
Intercept 281.49376639385366  
Prediction\_local [281.1144509]  
Right: 281.07794  
Intercept 281.5155053756187  
Prediction\_local [281.03447991]  
Right: 280.99893  
Intercept 281.50371307706473  
Prediction\_local [281.2479551]  
Right: 280.80432  
Intercept 281.38781526246214  
Prediction\_local [281.33727796]  
Right: 280.98022  
Intercept 281.4054354382302  
Prediction\_local [281.372018]  
Right: 281.3917  
Intercept 281.3486219286871  
Prediction\_local [281.47884574]  
Right: 281.64032  
Intercept 281.3212317146742  
Prediction\_local [281.74246667]  
Right: 281.91147  
Intercept 281.27597758032863  
Prediction\_local [281.80320041]  
Right: 281.9318  
Intercept 281.3302687351272  
Prediction\_local [281.78763542]  
Right: 281.838  
Intercept 281.2740656902128  
Prediction\_local [281.48917773]  
Right: 281.5718  
Intercept 281.3169115817664  
Prediction\_local [281.61918649]  
Right: 281.3801  
Intercept 281.39861789043  
Prediction\_local [281.26744464]  
Right: 281.17688  
Intercept 281.4859963702825  
Prediction\_local [281.07589519]  
Right: 280.9631  
Intercept 281.4057341063579  
Prediction\_local [281.28175941]  
Right: 280.85046  
Intercept 281.5265421538149  
Prediction\_local [281.21134784]  
Right: 280.7035  
Intercept 281.41306685920944  
Prediction\_local [281.33633524]  
Right: 280.941  
Intercept 281.44401301913877  
Prediction\_local [281.27368449]  
Right: 281.31107  
Intercept 281.38124681418213  
Prediction\_local [281.60293362]  
Right: 281.6403  
Intercept 281.3322761243733  
Prediction\_local [281.53599245]  
Right: 281.82986  
Intercept 281.2852036153352  
Prediction\_local [281.80282085]  
Right: 281.95694  
Intercept 281.3379983864278  
Prediction\_local [281.87011402]  
Right: 281.88086  
Intercept 281.3948215453447  
Prediction\_local [281.55104748]  
Right: 281.62866  
Intercept 281.3257185500772  
Prediction\_local [281.53641003]  
Right: 281.4271  
Intercept 281.4202611354204  
Prediction\_local [281.38750338]  
Right: 281.22394  
Intercept 281.47587581465865  
Prediction\_local [281.29012338]  
Right: 281.01202  
Intercept 281.5121551722163  
Prediction\_local [281.08189292]  
Right: 280.84006  
Intercept 281.46094170145733  
Prediction\_local [281.34191369]  
Right: 280.7111  
Intercept 281.4663593960088  
Prediction\_local [281.20453166]  
Right: 280.73822  
Intercept 281.46316220865623  
Prediction\_local [281.22287746]  
Right: 281.10214  
Intercept 281.5317227812416  
Prediction\_local [281.22546046]  
Right: 281.38644  
Intercept 281.3815965621476  
Prediction\_local [281.38880904]  
Right: 281.6687  
Intercept 281.27905580748  
Prediction\_local [281.82262647]  
Right: 281.74686  
Intercept 281.24281390728356  
Prediction\_local [281.85976627]  
Right: 281.6582  
Intercept 281.33561682226934  
Prediction\_local [281.51728943]  
Right: 281.46497  
Intercept 281.3493755037011  
Prediction\_local [281.55895602]  
Right: 281.19864  
Intercept 281.44216331433364  
Prediction\_local [281.1249451]  
Right: 281.10822  
Intercept 281.4971590889431  
Prediction\_local [281.12712676]  
Right: 280.92227  
Intercept 281.48031328838505  
Prediction\_local [281.10141201]  
Right: 280.78107  
Intercept 281.55048890273525  
Prediction\_local [281.01631818]  
Right: 280.5763  
Intercept 281.44303408406046  
Prediction\_local [281.12826391]  
Right: 280.74533  
Intercept 281.41838303197284  
Prediction\_local [281.35011238]  
Right: 281.30612  
Intercept 281.42634647568786  
Prediction\_local [281.15746965]  
Right: 281.70407  
Intercept 281.31402485361195  
Prediction\_local [281.41223686]  
Right: 281.8118  
Intercept 281.3202747826733  
Prediction\_local [281.85617502]  
Right: 281.91882  
Intercept 281.25536576834304  
Prediction\_local [281.92844443]  
Right: 281.7381  
Intercept 281.3656051599843  
Prediction\_local [281.5228829]  
Right: 281.49493  
Intercept 281.3228319740041  
Prediction\_local [281.70654547]  
Right: 281.3149  
Intercept 281.3476393945477  
Prediction\_local [281.51529168]  
Right: 281.1655  
Intercept 281.5385555110955  
Prediction\_local [280.97450506]  
Right: 281.0  
Intercept 281.4743955989929  
Prediction\_local [281.11372382]  
Right: 280.77444  
Intercept 281.54767425796007  
Prediction\_local [280.96217645]  
Right: 280.65298  
Intercept 281.5386154652534  
Prediction\_local [280.9987865]  
Right: 280.93628  
Intercept 281.4928213688606  
Prediction\_local [281.09145932]  
Right: 281.14578  
Intercept 281.5154896582975  
Prediction\_local [280.98197863]  
Right: 281.65616  
Intercept 281.38960098070044  
Prediction\_local [281.27509574]  
Right: 281.97482  
Intercept 281.24244306254485  
Prediction\_local [281.81133071]  
Right: 282.1044  
Intercept 281.32450536210195  
Prediction\_local [281.70370274]  
Right: 282.08075  
Intercept 281.3552555724909  
Prediction\_local [281.5904339]  
Right: 281.7801  
Intercept 281.34508271294504  
Prediction\_local [281.62540285]  
Right: 281.41776  
Intercept 281.4630228719697  
Prediction\_local [281.28235709]  
Right: 281.25943  
Intercept 281.49613694220784  
Prediction\_local [281.16437254]  
Right: 280.99622  
Intercept 281.54210978456206  
Prediction\_local [281.25286102]  
Right: 280.79794  
Intercept 281.57285868624524  
Prediction\_local [280.91015745]  
Right: 280.74646  
Intercept 281.44843536791404  
Prediction\_local [281.1270265]  
Right: 280.8895  
Intercept 281.46116698036667  
Prediction\_local [281.33877451]  
Right: 281.27753  
Intercept 281.41653229541316  
Prediction\_local [281.16118141]  
Right: 281.6057  
Intercept 281.4603884947007  
Prediction\_local [281.22681698]  
Right: 281.7855  
Intercept 281.31817406870795  
Prediction\_local [281.83675974]  
Right: 282.01523  
Intercept 281.2298571323408  
Prediction\_local [281.89953268]  
Right: 281.86462  
Intercept 281.3859844976068  
Prediction\_local [281.44237355]  
Right: 281.56702  
Intercept 281.30331994246575  
Prediction\_local [281.67394034]  
Right: 281.43854  
Intercept 281.40760923113237  
Prediction\_local [281.30492163]  
Right: 281.3645  
Intercept 281.4942015736244  
Prediction\_local [281.05036805]  
Right: 281.23987  
Intercept 281.51818816690417  
Prediction\_local [280.97667658]  
Right: 280.84198  
Intercept 281.5077279056396  
Prediction\_local [280.99150968]  
Right: 280.96704  
Intercept 281.5172666684009  
Prediction\_local [281.15286747]  
Right: 281.06027  
Intercept 281.4149943260356  
Prediction\_local [281.31903661]  
Right: 281.39703  
Intercept 281.43492765195595  
Prediction\_local [281.3089412]  
Right: 281.78802  
Intercept 281.4475478347238  
Prediction\_local [281.27202701]  
Right: 282.11508  
Intercept 281.24052452585977  
Prediction\_local [281.69020873]  
Right: 282.2937  
Intercept 281.22320734991746  
Prediction\_local [281.87722948]  
Right: 282.24976  
Intercept 281.3350046002439  
Prediction\_local [281.57599624]  
Right: 281.96033  
Intercept 281.2951801284901  
Prediction\_local [281.81134769]  
Right: 281.80402  
Intercept 281.37957999523826  
Prediction\_local [281.38506097]  
Right: 281.6671  
Intercept 281.4476946832418  
Prediction\_local [281.23405378]  
Right: 281.4137  
Intercept 281.4758184495794  
Prediction\_local [281.28341666]  
Right: 281.24316  
Intercept 281.5493524058204  
Prediction\_local [281.08437408]  
Right: 281.14365  
Intercept 281.455564321836  
Prediction\_local [281.24920551]  
Right: 281.21072  
Intercept 281.4559812433149  
Prediction\_local [281.33486253]  
Right: 281.49695  
Intercept 281.4947214965245  
Prediction\_local [281.3383519]  
Right: 281.80014  
Intercept 281.4417951665882  
Prediction\_local [281.27563667]  
Right: 282.04675  
Intercept 281.25260437581176  
Prediction\_local [281.87743333]  
Right: 282.1892  
Intercept 281.2251727212611  
Prediction\_local [281.86482831]  
Right: 282.11188  
Intercept 281.3129327609078  
Prediction\_local [281.62477689]  
Right: 281.97678  
Intercept 281.3534411736333  
Prediction\_local [281.66191669]  
Right: 281.7694  
Intercept 281.3217287347119  
Prediction\_local [281.62422521]  
Right: 281.46567  
Intercept 281.4404409296164  
Prediction\_local [281.14428113]  
Right: 281.22458  
Intercept 281.43276223755424  
Prediction\_local [281.25805742]  
Right: 281.13788  
Intercept 281.37229631147903  
Prediction\_local [281.40565357]  
Right: 281.02768  
Intercept 281.4356103765932  
Prediction\_local [281.34418414]  
Right: 280.9718  
Intercept 281.4408057699564  
Prediction\_local [281.47683484]  
Right: 281.46985  
Intercept 281.2775054260254  
Prediction\_local [281.53630659]  
Right: 281.868  
Intercept 281.34585169547194  
Prediction\_local [281.5226119]  
Right: 282.01086  
Intercept 281.2927742044457  
Prediction\_local [281.7959884]  
Right: 281.97183  
Intercept 281.2995415038057  
Prediction\_local [281.71496401]  
Right: 281.79813

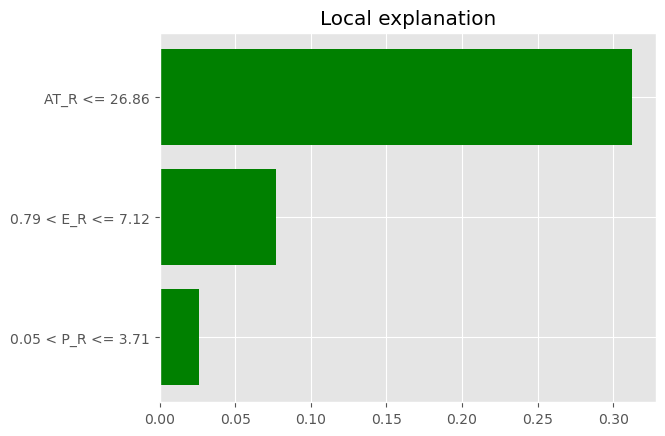
exp

<lime.explanation.Explanation at 0x18b609f1f40>

##### Showing the explanable table  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('AT\_R <= 26.86', 0.31254071694051755),  
 ('0.79 < E\_R <= 7.12', 0.07711463997290209),  
 ('0.05 < P\_R <= 3.71', 0.02576715230325543)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.31254071694051755),  
 (0, -0.07711463997290209),  
 (2, -0.02576715230325543)],  
 1: [(1, 0.31254071694051755),  
 (0, 0.07711463997290209),  
 (2, 0.02576715230325543)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.71496401]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.79813

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb1RR\_Whole.html')

####################################################################################################

#The best parameters are: {'colsample\_bytree': 0.7, 'learning\_rate': 0.1, 'max\_depth': 9, 'n\_estimators': 100, 'subsample': 0.5}  
#The best score is: -0.12347843485644353

########### Fitting the model on the training dataset using the best parameters  
model = XGBRegressor()  
xgb\_param\_grid = {'colsample\_bytree' : [0.7],  
 'learning\_rate' : [0.1],  
 'max\_depth' : [9],  
 'n\_estimators' : [100],  
 'subsample' : [0.5],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb1 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb1.fit(X\_R\_train, y\_R\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 max\_cat\_to\_onehot=None, max\_delta\_step=None,  
 max\_depth=None, max\_leaves=None,  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [0.7], 'learning\_rate': [0.1],  
 'max\_depth': [9], 'n\_estimators': [100],  
 'subsample': [0.5]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The best score is:', xgb1.best\_score\_)

The best score is: -0.1610004524583158

###########################################################

# Initialization  
model = XGBRegressor(colsample\_bytree = 0.7, learning\_rate = 0.1, max\_depth = 9, n\_estimators = 100, subsample = 0.5)

# Fitting the model  
xgb1 = model.fit(X\_R\_train, y\_R\_train)

# Evaluation of the performance of the model on the training dataset:  
xgb\_y\_predtr = xgb1.predict(X\_R\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_R\_train, xgb\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_R\_train, xgb\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_R\_train, xgb\_y\_predtr))

The training MAE is: 0.032168240017358726  
The training MSE is: 0.002014634969352004  
The training R2 Score is: 0.9904638453933845

# Evaluation of the performance of the model on the testing dataset:  
xgb\_y\_predts = xgb1.predict(X\_R\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_R\_test, xgb\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_R\_test, xgb\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_R\_test, xgb\_y\_predts))

The testing MAE is: 0.25260073061341853  
The testing MSE is: 0.10919470082841017  
The testing R2 Score is: 0.2482004991923984

########### Cross-validation ###########################

############# On the training dataset  
   
score\_gb1tr = cross\_val\_score(xgb1, X\_R\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_gb1tr)))

The mean training CV score is: 0.17811311982218175

############# On the testing dataset   
score\_gb1ts = cross\_val\_score(xgb1, X\_R\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_gb1ts)))

The mean testing CV score is: 0.08635062942492519

################ LIME ################

###### Initializing the explainer on the training dataset  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, xgb1.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.4588267194348  
Prediction\_local [281.24788628]  
Right: 281.36996  
Intercept 281.4053908591422  
Prediction\_local [281.18324431]  
Right: 281.22073  
Intercept 281.32813350146995  
Prediction\_local [281.61329579]  
Right: 282.05182  
Intercept 281.35080201675373  
Prediction\_local [281.24337212]  
Right: 280.77472  
Intercept 281.4017876495239  
Prediction\_local [281.26523746]  
Right: 281.07748  
Intercept 281.21993251278843  
Prediction\_local [281.75959234]  
Right: 281.9269  
Intercept 281.444135719164  
Prediction\_local [281.15529056]  
Right: 281.1819  
Intercept 281.4497655530236  
Prediction\_local [281.01469573]  
Right: 280.99304  
Intercept 281.43191041155063  
Prediction\_local [281.11839121]  
Right: 280.76126  
Intercept 281.4439538785755  
Prediction\_local [281.12683338]  
Right: 281.04495  
Intercept 281.45149396593496  
Prediction\_local [281.12065303]  
Right: 281.75436  
Intercept 281.3975492342756  
Prediction\_local [281.08337884]  
Right: 280.96  
Intercept 281.25557440468623  
Prediction\_local [281.48964134]  
Right: 281.67566  
Intercept 281.42109624718506  
Prediction\_local [281.34865354]  
Right: 281.46432  
Intercept 281.2183222133457  
Prediction\_local [281.60069726]  
Right: 281.85437  
Intercept 281.47056772743986  
Prediction\_local [280.99581599]  
Right: 280.959  
Intercept 281.2099689425994  
Prediction\_local [281.68204059]  
Right: 281.7723  
Intercept 281.3308435199889  
Prediction\_local [281.23885534]  
Right: 281.61792  
Intercept 281.42501892940317  
Prediction\_local [281.06879517]  
Right: 280.78143  
Intercept 281.43166567324096  
Prediction\_local [281.20717758]  
Right: 280.788  
Intercept 281.28280399256107  
Prediction\_local [281.33966893]  
Right: 281.31573  
Intercept 281.27264012415776  
Prediction\_local [281.78566336]  
Right: 282.20132  
Intercept 281.3651269125985  
Prediction\_local [281.25878511]  
Right: 281.18103  
Intercept 281.44226030519366  
Prediction\_local [280.91833937]  
Right: 281.0241  
Intercept 281.32437571116014  
Prediction\_local [281.21775815]  
Right: 281.2295  
Intercept 281.3327502355889  
Prediction\_local [281.49693458]  
Right: 281.92654  
Intercept 281.45528502154104  
Prediction\_local [281.06650039]  
Right: 280.73972  
Intercept 281.26872644184044  
Prediction\_local [281.54597194]  
Right: 281.5252  
Intercept 281.49826572813225  
Prediction\_local [280.98986148]  
Right: 280.93817  
Intercept 281.269340737453  
Prediction\_local [281.60694361]  
Right: 281.41696  
Intercept 281.4099860809191  
Prediction\_local [280.9480743]  
Right: 280.76877  
Intercept 281.2302729246546  
Prediction\_local [281.70365805]  
Right: 281.96506  
Intercept 281.2598626264738  
Prediction\_local [281.59033752]  
Right: 281.4766  
Intercept 281.22147701695116  
Prediction\_local [281.75591757]  
Right: 282.14944  
Intercept 281.2358557160457  
Prediction\_local [281.57808177]  
Right: 281.72992  
Intercept 281.5315207362735  
Prediction\_local [280.96558526]  
Right: 281.59323  
Intercept 281.2679822177052  
Prediction\_local [281.56442541]  
Right: 281.40442  
Intercept 281.11177954418724  
Prediction\_local [281.79071047]  
Right: 281.86066  
Intercept 281.4570684372848  
Prediction\_local [281.0658326]  
Right: 280.9369  
Intercept 281.4030477131075  
Prediction\_local [281.12527339]  
Right: 281.8357  
Intercept 281.2461715403928  
Prediction\_local [281.75870276]  
Right: 281.96246  
Intercept 281.41498597614043  
Prediction\_local [281.33763546]  
Right: 281.6944  
Intercept 281.30997178898326  
Prediction\_local [281.60732319]  
Right: 281.9846  
Intercept 281.2392490108578  
Prediction\_local [281.59333951]  
Right: 282.08624  
Intercept 281.4454473881091  
Prediction\_local [280.91242037]  
Right: 280.8571  
Intercept 281.47506195517053  
Prediction\_local [281.22143084]  
Right: 281.14923  
Intercept 281.31175772813265  
Prediction\_local [281.48753516]  
Right: 281.67136  
Intercept 281.35630438296096  
Prediction\_local [281.61872483]  
Right: 281.17783  
Intercept 281.30951349359884  
Prediction\_local [281.64955858]  
Right: 281.3707  
Intercept 281.2447453272406  
Prediction\_local [281.32723779]  
Right: 280.9454  
Intercept 281.3985596967007  
Prediction\_local [281.17180704]  
Right: 280.97308  
Intercept 281.2997823354337  
Prediction\_local [281.56316275]  
Right: 281.86005  
Intercept 281.33129305705745  
Prediction\_local [281.49086061]  
Right: 282.02347  
Intercept 281.41024269972553  
Prediction\_local [280.98653099]  
Right: 280.80896  
Intercept 281.35020528192393  
Prediction\_local [281.47687808]  
Right: 281.80554  
Intercept 281.2199192362105  
Prediction\_local [281.69578289]  
Right: 281.46112  
Intercept 281.40716902157646  
Prediction\_local [281.32731681]  
Right: 280.98535  
Intercept 281.3806396999506  
Prediction\_local [281.04566345]  
Right: 281.27332  
Intercept 281.3800817461105  
Prediction\_local [281.38201743]  
Right: 280.97086  
Intercept 281.4134668765011  
Prediction\_local [281.02411381]  
Right: 280.9044  
Intercept 281.49892318173346  
Prediction\_local [281.02322831]  
Right: 280.80136  
Intercept 281.49276347575386  
Prediction\_local [280.97277333]  
Right: 281.17694  
Intercept 281.40985323062097  
Prediction\_local [281.22048054]  
Right: 280.89496  
Intercept 281.2281500988625  
Prediction\_local [281.76425686]  
Right: 281.67017  
Intercept 281.2422640367268  
Prediction\_local [281.56233154]  
Right: 281.42493  
Intercept 281.4375827898224  
Prediction\_local [281.57308299]  
Right: 281.83923  
Intercept 281.3030806946885  
Prediction\_local [281.55364027]  
Right: 281.2765  
Intercept 281.4717383120676  
Prediction\_local [280.92251194]  
Right: 280.73077  
Intercept 281.3474954403183  
Prediction\_local [281.34011921]  
Right: 281.38608  
Intercept 281.51162885211187  
Prediction\_local [280.89391653]  
Right: 280.64484  
Intercept 281.3898005872534  
Prediction\_local [281.27466438]  
Right: 281.10092  
Intercept 281.2550131626488  
Prediction\_local [281.70966106]  
Right: 281.91855  
Intercept 281.4598798020915  
Prediction\_local [281.06828076]  
Right: 280.70337  
Intercept 281.49260819845244  
Prediction\_local [280.88237683]  
Right: 280.6847  
Intercept 281.4072210224137  
Prediction\_local [281.30122902]  
Right: 281.3194  
Intercept 281.25636474406633  
Prediction\_local [281.78166969]  
Right: 281.90125  
Intercept 281.31242890922675  
Prediction\_local [281.61329898]  
Right: 281.59836  
Intercept 281.4110156783412  
Prediction\_local [281.25794096]  
Right: 281.89905  
Intercept 281.33837854740494  
Prediction\_local [281.30714111]  
Right: 281.86035  
Intercept 281.27348350642046  
Prediction\_local [281.58927148]  
Right: 281.793  
Intercept 281.27323849179476  
Prediction\_local [281.48791764]  
Right: 281.2284

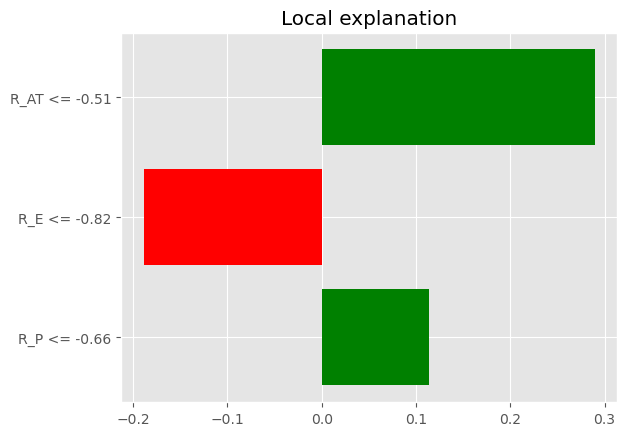
exp

<lime.explanation.Explanation at 0x25d944b6520>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_AT <= -0.51', 0.28969612457675553),  
 ('R\_E <= -0.82', -0.18850892097681704),  
 ('R\_P <= -0.66', 0.11349194433711285)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.28969612457675553),  
 (0, 0.18850892097681704),  
 (2, -0.11349194433711285)],  
 1: [(1, 0.28969612457675553),  
 (0, -0.18850892097681704),  
 (2, 0.11349194433711285)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.48791764]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.2284

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb1RR\_train.html')

################################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, xgb1.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 281.28796766562436  
Prediction\_local [281.57908452]  
Right: 281.5984  
Intercept 281.24395710854793  
Prediction\_local [281.5953664]  
Right: 281.9899  
Intercept 281.38961056535135  
Prediction\_local [281.10397051]  
Right: 281.2151  
Intercept 281.40323048907715  
Prediction\_local [281.51143476]  
Right: 281.39166  
Intercept 281.37968916425524  
Prediction\_local [281.08300344]  
Right: 281.27866  
Intercept 281.36264182430324  
Prediction\_local [281.36073226]  
Right: 281.1855  
Intercept 281.3383093059747  
Prediction\_local [281.56561158]  
Right: 281.53146  
Intercept 281.3231172110492  
Prediction\_local [281.47062488]  
Right: 281.5428  
Intercept 281.30114761336154  
Prediction\_local [281.33629344]  
Right: 281.5065  
Intercept 281.1909774050643  
Prediction\_local [281.52202614]  
Right: 281.92633  
Intercept 281.48533671331677  
Prediction\_local [281.01004412]  
Right: 280.93427  
Intercept 281.47115200374867  
Prediction\_local [281.31364225]  
Right: 281.89908  
Intercept 281.3954721714761  
Prediction\_local [281.35638422]  
Right: 281.38052  
Intercept 281.2656704722098  
Prediction\_local [281.37009342]  
Right: 281.3671  
Intercept 281.3227388348744  
Prediction\_local [281.15329933]  
Right: 281.10388  
Intercept 281.5052075351444  
Prediction\_local [280.93422323]  
Right: 280.8941  
Intercept 281.48838994805277  
Prediction\_local [281.27877532]  
Right: 281.57828  
Intercept 281.3942248534997  
Prediction\_local [281.44428454]  
Right: 281.62894  
Intercept 281.2544546007718  
Prediction\_local [281.44811331]  
Right: 281.1949  
Intercept 281.48647171682484  
Prediction\_local [281.27459784]  
Right: 281.2046  
Intercept 281.25711727914  
Prediction\_local [281.73612264]  
Right: 281.96835  
Intercept 281.49027157727903  
Prediction\_local [280.96583388]  
Right: 280.9622  
Intercept 281.4847484811308  
Prediction\_local [281.51393137]  
Right: 281.48807  
Intercept 281.2905351640688  
Prediction\_local [281.55855233]  
Right: 281.40762  
Intercept 281.2188084345941  
Prediction\_local [281.58125266]  
Right: 281.97357  
Intercept 281.4778449432753  
Prediction\_local [281.10027356]  
Right: 281.33246  
Intercept 281.2674613584521  
Prediction\_local [281.44201075]  
Right: 281.32162

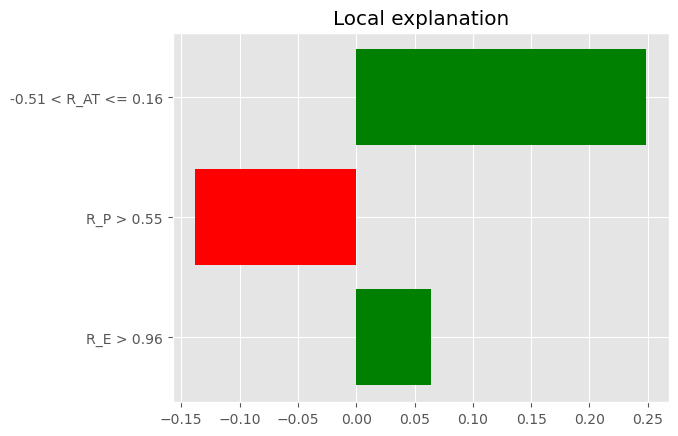
exp

<lime.explanation.Explanation at 0x25d95a69910>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('-0.51 < R\_AT <= 0.16', 0.24871322331136453),  
 ('R\_P > 0.55', -0.13787883673794973),  
 ('R\_E > 0.96', 0.06371500759197621)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.24871322331136453),  
 (2, 0.13787883673794973),  
 (0, -0.06371500759197621)],  
 1: [(1, 0.24871322331136453),  
 (2, -0.13787883673794973),  
 (0, 0.06371500759197621)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.44201075]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.32162

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb1RR\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

############ MODEL 2: GT Features (X\_G) and RS Target (LL\_R) ###############

########### On the whole dataset  
model = XGBRegressor()  
xgb\_param\_grid = { 'colsample\_bytree' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'learning\_rate' : [0.001, 0.1, 0.20, 0.30],  
 'max\_depth' : [3, 6, 9, 12],  
 'n\_estimators' : [100, 500, 1000],  
 'subsample' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb2 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb2.fit(X\_G, y\_R)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'learning\_rate': [0.001, 0.1, 0.2, 0.3],  
 'max\_depth': [3, 6, 9, 12],  
 'n\_estimators': [100, 500, 1000],  
 'subsample': [0.1, 0.3, 0.5, 0.7, 0.9, 1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The best parameters are:', xgb2.best\_params\_)  
print('The best score is:', xgb2.best\_score\_)

The best parameters are: {'colsample\_bytree': 0.7, 'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100, 'subsample': 0.9}  
The best score is: -0.07112854410434075

#The best parameters are: {'colsample\_bytree': 0.7, 'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100, 'subsample': 0.9}  
#The best score is: -0.07112854410434075

# Evaluation of the performance of the regression model on the whole dataset:  
xgb\_y\_pred = xgb2.predict(X\_G)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_R, xgb\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_R, xgb\_y\_pred))  
print('R2 Score:', r2\_score(y\_R, xgb\_y\_pred))

Mean Absolute Error: 0.08133373119212897  
Mean Squared Error: 0.010278703483138509  
R2 Score: 0.9496337824263912

#####################LIME ##############################

###### Initializing the explainer on the whole datset  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the whole XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, xgb2.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 281.39033752714977  
Prediction\_local [281.88768393]  
Right: 281.83298  
Intercept 281.3361680200095  
Prediction\_local [281.78712553]  
Right: 281.5846  
Intercept 281.6094789152696  
Prediction\_local [281.06320056]  
Right: 281.2446  
Intercept 281.64667589214  
Prediction\_local [281.06967328]  
Right: 281.0014  
Intercept 281.5408371880507  
Prediction\_local [281.02366213]  
Right: 280.9177  
Intercept 281.6451063008131  
Prediction\_local [281.16527997]  
Right: 280.83957  
Intercept 281.55954258637536  
Prediction\_local [281.38495631]  
Right: 281.01764  
Intercept 281.4881742249734  
Prediction\_local [281.39689459]  
Right: 281.28128  
Intercept 281.40302669188895  
Prediction\_local [281.99349607]  
Right: 281.70673  
Intercept 281.3245652101951  
Prediction\_local [281.91679094]  
Right: 281.89853  
Intercept 281.4173967907614  
Prediction\_local [281.87081685]  
Right: 281.96066  
Intercept 281.3735258760728  
Prediction\_local [281.75252324]  
Right: 281.89536  
Intercept 281.3612870917091  
Prediction\_local [281.89162645]  
Right: 281.5993  
Intercept 281.4424569243276  
Prediction\_local [281.69179372]  
Right: 281.56958  
Intercept 281.5169645530053  
Prediction\_local [281.36425692]  
Right: 281.33817  
Intercept 281.64257737162734  
Prediction\_local [280.97097346]  
Right: 281.0958  
Intercept 281.63102549061495  
Prediction\_local [281.13196411]  
Right: 280.94  
Intercept 281.6265670258097  
Prediction\_local [281.12464999]  
Right: 280.89844  
Intercept 281.68204592622425  
Prediction\_local [281.12049498]  
Right: 280.88293  
Intercept 281.5463565079589  
Prediction\_local [281.36499341]  
Right: 281.2567  
Intercept 281.5400831126893  
Prediction\_local [281.53507502]  
Right: 281.5772  
Intercept 281.39032368422784  
Prediction\_local [281.8358524]  
Right: 281.86478  
Intercept 281.3266778938077  
Prediction\_local [281.87249742]  
Right: 281.96597  
Intercept 281.3184644056337  
Prediction\_local [281.86688814]  
Right: 281.83868  
Intercept 281.30365120130034  
Prediction\_local [281.86320582]  
Right: 281.61844  
Intercept 281.4012663929219  
Prediction\_local [281.67913871]  
Right: 281.49213  
Intercept 281.70551442322676  
Prediction\_local [280.8934152]  
Right: 281.25308  
Intercept 281.5912820312299  
Prediction\_local [281.14907407]  
Right: 280.9509  
Intercept 281.62533830575  
Prediction\_local [281.00108606]  
Right: 280.9177  
Intercept 281.66487579818954  
Prediction\_local [280.98592197]  
Right: 280.67856  
Intercept 281.4585616494332  
Prediction\_local [281.61001223]  
Right: 280.92795  
Intercept 281.49769446316515  
Prediction\_local [281.61613719]  
Right: 281.15976  
Intercept 281.4402462329055  
Prediction\_local [281.88114102]  
Right: 281.34293  
Intercept 281.44999077699174  
Prediction\_local [281.7928569]  
Right: 281.81842  
Intercept 281.3282677218894  
Prediction\_local [281.85931245]  
Right: 281.64627  
Intercept 281.36980943451294  
Prediction\_local [281.76426288]  
Right: 281.58466  
Intercept 281.4576870412658  
Prediction\_local [281.63866845]  
Right: 281.41486  
Intercept 281.52694381201655  
Prediction\_local [281.54525196]  
Right: 281.2675  
Intercept 281.47939872989014  
Prediction\_local [281.31859953]  
Right: 281.19934  
Intercept 281.60412234254704  
Prediction\_local [281.09500382]  
Right: 280.9862  
Intercept 281.6894936007049  
Prediction\_local [281.02909388]  
Right: 280.7675  
Intercept 281.65668227048496  
Prediction\_local [281.28834877]  
Right: 280.63345  
Intercept 281.51740963177167  
Prediction\_local [281.37636113]  
Right: 280.88474  
Intercept 281.4166807019357  
Prediction\_local [281.78004907]  
Right: 281.2987  
Intercept 281.4255176469882  
Prediction\_local [281.83177115]  
Right: 281.74207  
Intercept 281.48279195451346  
Prediction\_local [281.49251519]  
Right: 281.86322  
Intercept 281.3607641578383  
Prediction\_local [281.86474481]  
Right: 281.95898  
Intercept 281.42999023592193  
Prediction\_local [281.86223679]  
Right: 281.85544  
Intercept 281.4279874372836  
Prediction\_local [281.69909032]  
Right: 281.50357  
Intercept 281.4161760310202  
Prediction\_local [281.67555768]  
Right: 281.41632  
Intercept 281.40745374565853  
Prediction\_local [281.62525112]  
Right: 281.3248  
Intercept 281.710322143663  
Prediction\_local [281.01598497]  
Right: 281.03555  
Intercept 281.680380769833  
Prediction\_local [281.08433448]  
Right: 280.9136  
Intercept 281.570299330983  
Prediction\_local [281.07033609]  
Right: 280.76447  
Intercept 281.55639055547607  
Prediction\_local [281.39254936]  
Right: 280.928  
Intercept 281.5170131728909  
Prediction\_local [281.54304676]  
Right: 281.06287  
Intercept 281.50588276656026  
Prediction\_local [281.58119039]  
Right: 281.56415  
Intercept 281.5294548161665  
Prediction\_local [281.33513852]  
Right: 281.91434  
Intercept 281.2154410178495  
Prediction\_local [281.90036186]  
Right: 282.0848  
Intercept 281.3433665241437  
Prediction\_local [281.89502353]  
Right: 282.02048  
Intercept 281.4295313860401  
Prediction\_local [281.70085897]  
Right: 281.7161  
Intercept 281.43844030887306  
Prediction\_local [281.68269285]  
Right: 281.47266  
Intercept 281.519703880134  
Prediction\_local [281.29908636]  
Right: 281.26984  
Intercept 281.6045290677474  
Prediction\_local [281.12185834]  
Right: 281.03745  
Intercept 281.5985587062718  
Prediction\_local [281.1157621]  
Right: 280.81384  
Intercept 281.67221005160576  
Prediction\_local [281.05325016]  
Right: 280.83826  
Intercept 281.5818048616782  
Prediction\_local [281.26610692]  
Right: 280.97934  
Intercept 281.3209120544588  
Prediction\_local [281.7634434]  
Right: 281.33035  
Intercept 281.4310583841758  
Prediction\_local [281.690967]  
Right: 281.52213  
Intercept 281.4164263140585  
Prediction\_local [281.92224817]  
Right: 281.9198  
Intercept 281.3487585866237  
Prediction\_local [281.78195882]  
Right: 282.06036  
Intercept 281.37238974939373  
Prediction\_local [281.93290982]  
Right: 281.87897  
Intercept 281.3773443580313  
Prediction\_local [281.69961933]  
Right: 281.59756  
Intercept 281.44184473543936  
Prediction\_local [281.69977729]  
Right: 281.47595  
Intercept 281.5372981463795  
Prediction\_local [281.27956425]  
Right: 281.28677  
Intercept 281.600452810323  
Prediction\_local [281.04339278]  
Right: 281.1052  
Intercept 281.5615952510909  
Prediction\_local [281.06264816]  
Right: 280.89102  
Intercept 281.6497600531194  
Prediction\_local [280.88982493]  
Right: 280.87738  
Intercept 281.603193867108  
Prediction\_local [281.33093987]  
Right: 280.96768  
Intercept 281.58464121002856  
Prediction\_local [281.37421086]  
Right: 281.41803  
Intercept 281.53474069422174  
Prediction\_local [281.45053963]  
Right: 281.79437  
Intercept 281.47588775115605  
Prediction\_local [281.73067126]  
Right: 282.163  
Intercept 281.3566468350572  
Prediction\_local [281.99842173]  
Right: 282.15622  
Intercept 281.2722130196702  
Prediction\_local [281.83099721]  
Right: 282.03583  
Intercept 281.39193360727876  
Prediction\_local [281.75225737]  
Right: 281.95316  
Intercept 281.4318660992754  
Prediction\_local [281.75011909]  
Right: 281.68637  
Intercept 281.6360572698649  
Prediction\_local [281.31021172]  
Right: 281.50592  
Intercept 281.68412064849105  
Prediction\_local [281.05851015]  
Right: 281.29633  
Intercept 281.54848292514464  
Prediction\_local [281.12628138]  
Right: 281.0525  
Intercept 281.7092080098734  
Prediction\_local [280.99464215]  
Right: 281.0161  
Intercept 281.56815052924946  
Prediction\_local [281.34778899]  
Right: 281.13083  
Intercept 281.59273027191585  
Prediction\_local [281.48095756]  
Right: 281.43317  
Intercept 281.3920740323757  
Prediction\_local [281.69453463]  
Right: 281.82993  
Intercept 281.42129151454634  
Prediction\_local [281.95039707]  
Right: 281.98917  
Intercept 281.26682342487953  
Prediction\_local [281.95150813]  
Right: 282.13586  
Intercept 281.2398122941146  
Prediction\_local [281.956276]  
Right: 282.10883  
Intercept 281.45083786079834  
Prediction\_local [281.6753486]  
Right: 281.84808  
Intercept 281.5499794530922  
Prediction\_local [281.6248201]  
Right: 281.72552  
Intercept 281.42493405298774  
Prediction\_local [281.48936745]  
Right: 281.33606  
Intercept 281.61642982647123  
Prediction\_local [281.03139371]  
Right: 281.12482  
Intercept 281.6431945173729  
Prediction\_local [281.11668424]  
Right: 281.13315  
Intercept 281.56151364819783  
Prediction\_local [281.42204754]  
Right: 281.0843  
Intercept 281.58307273988555  
Prediction\_local [281.51700574]  
Right: 281.12207  
Intercept 281.48284653148204  
Prediction\_local [281.4350144]  
Right: 281.40567  
Intercept 281.47238439459676  
Prediction\_local [281.70235087]  
Right: 281.67612  
Intercept 281.4948327041399  
Prediction\_local [281.79564844]  
Right: 281.8581  
Intercept 281.37488915531316  
Prediction\_local [281.95582784]  
Right: 282.04968  
Intercept 281.3861933393669  
Prediction\_local [281.77715937]  
Right: 281.95526

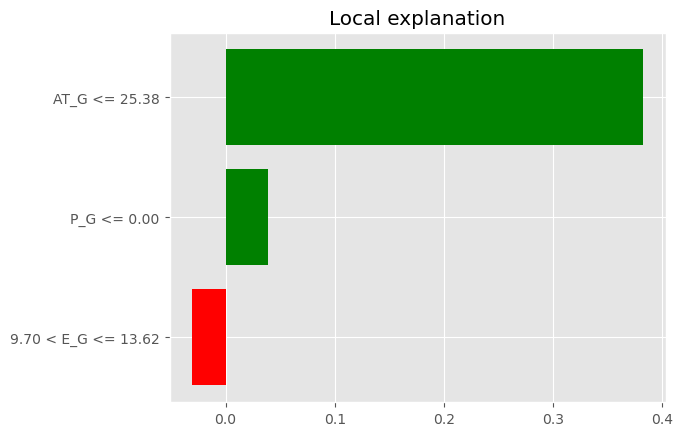
exp

<lime.explanation.Explanation at 0x18b616939d0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('AT\_G <= 25.38', 0.3822989985757926),  
 ('P\_G <= 0.00', 0.039289549109363456),  
 ('9.70 < E\_G <= 13.62', -0.03062251611403699)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.3822989985757926),  
 (2, -0.039289549109363456),  
 (0, 0.03062251611403699)],  
 1: [(1, 0.3822989985757926),  
 (2, 0.039289549109363456),  
 (0, -0.03062251611403699)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.77715937]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.95526

# Saving fe# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb2GR\_Whole.html')

##########################################################################################

#The best parameters are: {'colsample\_bytree': 0.7, 'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100, 'subsample': 0.9}  
#The best score is: -0.07112854410434075

########### Fitting the model on the training dataset using the best parameters  
model = XGBRegressor()  
xgb\_param\_grid = {'colsample\_bytree' : [0.7],  
 'learning\_rate' : [0.1],  
 'max\_depth' : [3],  
 'n\_estimators' : [100],  
 'subsample' : [0.9],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb2 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb2.fit(X\_G\_train, y\_R\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 max\_cat\_to\_onehot=None, max\_delta\_step=None,  
 max\_depth=None, max\_leaves=None,  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [0.7], 'learning\_rate': [0.1],  
 'max\_depth': [3], 'n\_estimators': [100],  
 'subsample': [0.9]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The best score is:', xgb2.best\_score\_)

The best score is: -0.08876829728522974

# Initialization  
model = XGBRegressor(colsample\_bytree = 0.7, learning\_rate = 0.1, max\_depth = 3, n\_estimators = 100, subsample = 0.9)

# Fitting the model  
xgb2 = model.fit(X\_G\_train, y\_R\_train)

# Evaluation of the performance of the model on the training dataset:  
xgb\_y\_predtr = xgb2.predict(X\_G\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, xgb\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, xgb\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, xgb\_y\_predtr))

The training MAE is: 1.6568462305893152  
The training MSE is: 2.829891146031122  
The training R2 Score is: -8.281492084531523

# Evaluation of the performance of the model on the testing dataset:  
xgb\_y\_predts = xgb2.predict(X\_G\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, xgb\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, xgb\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, xgb\_y\_predts))

The testing MAE is: 1.5641902669270829  
The testing MSE is: 2.616226004884313  
The testing R2 Score is: -8.540188827866322

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_xgb2tr = cross\_val\_score(xgb2, X\_G\_train, y\_R\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_xgb2tr)))

The mean training CV score is: 0.09059738366898692

############# On the testing dataset   
score\_xgb2ts = cross\_val\_score(xgb2, X\_G\_test, y\_R\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_xgb2ts)))

The mean testing CV score is: 0.09070098264602186

#####################LIME ##############################

###### Initializing the explainer on the training dataset  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_R'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, xgb2.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 281.4219599159543  
Prediction\_local [281.26119865]  
Right: 281.338  
Intercept 281.5203160348193  
Prediction\_local [280.99028301]  
Right: 281.10797  
Intercept 281.16598193822495  
Prediction\_local [281.8804088]  
Right: 281.9633  
Intercept 281.4852692728544  
Prediction\_local [281.17322067]  
Right: 280.87  
Intercept 281.2614579944286  
Prediction\_local [281.46867324]  
Right: 281.1626  
Intercept 281.1777698324943  
Prediction\_local [281.83093322]  
Right: 281.9608  
Intercept 281.372581299387  
Prediction\_local [281.46894522]  
Right: 281.15173  
Intercept 281.54064888099316  
Prediction\_local [281.01109127]  
Right: 280.8924  
Intercept 281.54405767985924  
Prediction\_local [281.04247187]  
Right: 280.7964  
Intercept 281.4510447897516  
Prediction\_local [281.21968041]  
Right: 281.06168  
Intercept 281.33417526739794  
Prediction\_local [281.87477584]  
Right: 281.80878  
Intercept 281.4681671380202  
Prediction\_local [281.05870034]  
Right: 280.95596  
Intercept 281.3908756521806  
Prediction\_local [281.25320381]  
Right: 281.54932  
Intercept 281.4404354521451  
Prediction\_local [281.19934449]  
Right: 281.42166  
Intercept 281.2506120986301  
Prediction\_local [281.85633149]  
Right: 281.90005  
Intercept 281.5145629991274  
Prediction\_local [280.89548391]  
Right: 280.84763  
Intercept 281.22897186128614  
Prediction\_local [281.81230998]  
Right: 281.6734  
Intercept 281.34512316477765  
Prediction\_local [281.55358286]  
Right: 281.52173  
Intercept 281.49433495703653  
Prediction\_local [281.17989888]  
Right: 280.67438  
Intercept 281.38212101643137  
Prediction\_local [281.39552111]  
Right: 280.8881  
Intercept 281.4914769893646  
Prediction\_local [281.06673169]  
Right: 281.28284  
Intercept 281.08561237991665  
Prediction\_local [281.86818684]  
Right: 282.13297  
Intercept 281.37941879524897  
Prediction\_local [281.32273035]  
Right: 281.3052  
Intercept 281.58280407941095  
Prediction\_local [280.93339345]  
Right: 280.99518  
Intercept 281.46184903473295  
Prediction\_local [281.12594625]  
Right: 281.2549  
Intercept 281.3604658525744  
Prediction\_local [281.62387917]  
Right: 281.8888  
Intercept 281.5405646488233  
Prediction\_local [281.02424629]  
Right: 280.87476  
Intercept 281.30635745089086  
Prediction\_local [281.59723776]  
Right: 281.4622  
Intercept 281.44038235809955  
Prediction\_local [281.100597]  
Right: 280.95175  
Intercept 281.291639916994  
Prediction\_local [281.64397857]  
Right: 281.50235  
Intercept 281.5310562203409  
Prediction\_local [281.09511866]  
Right: 280.81552  
Intercept 281.2359442235896  
Prediction\_local [281.72930233]  
Right: 281.96457  
Intercept 281.2812819650177  
Prediction\_local [281.61512533]  
Right: 281.40964  
Intercept 281.2147230938156  
Prediction\_local [281.87047971]  
Right: 282.06046  
Intercept 281.1822871929853  
Prediction\_local [281.82446683]  
Right: 281.87177  
Intercept 281.4673378899661  
Prediction\_local [281.39006917]  
Right: 281.53558  
Intercept 281.3315860539514  
Prediction\_local [281.53029344]  
Right: 281.4514  
Intercept 281.1612825907094  
Prediction\_local [281.78545373]  
Right: 281.91428  
Intercept 281.4564179270766  
Prediction\_local [281.09305823]  
Right: 280.95706  
Intercept 281.2772859688992  
Prediction\_local [281.85807949]  
Right: 281.9305  
Intercept 281.2090274704939  
Prediction\_local [281.82719226]  
Right: 281.9954  
Intercept 281.38156026070067  
Prediction\_local [281.66889874]  
Right: 281.7283  
Intercept 281.21052024854845  
Prediction\_local [281.80076008]  
Right: 281.93195  
Intercept 281.17052244749505  
Prediction\_local [281.78759014]  
Right: 282.0392  
Intercept 281.5518138612162  
Prediction\_local [280.98233387]  
Right: 280.8888  
Intercept 281.50568242733016  
Prediction\_local [281.16212277]  
Right: 281.1189  
Intercept 281.1616866756374  
Prediction\_local [281.85300474]  
Right: 281.79843  
Intercept 281.30014503838237  
Prediction\_local [281.55632663]  
Right: 281.2797  
Intercept 281.30178149921875  
Prediction\_local [281.54740466]  
Right: 281.49515  
Intercept 281.4629981854875  
Prediction\_local [281.16686313]  
Right: 281.03726  
Intercept 281.4708720232563  
Prediction\_local [280.97749207]  
Right: 281.0027  
Intercept 281.2729924769288  
Prediction\_local [281.88448793]  
Right: 281.87585  
Intercept 281.2104387440626  
Prediction\_local [281.63407845]  
Right: 281.9903  
Intercept 281.44273360714647  
Prediction\_local [281.07442146]  
Right: 280.80914  
Intercept 281.322465695237  
Prediction\_local [281.46625184]  
Right: 281.80374  
Intercept 281.3399492743397  
Prediction\_local [281.39534213]  
Right: 281.4209  
Intercept 281.5400638138535  
Prediction\_local [281.09404169]  
Right: 280.93005  
Intercept 281.45232635801494  
Prediction\_local [281.20551339]  
Right: 281.2572  
Intercept 281.4765091865807  
Prediction\_local [281.2574367]  
Right: 281.01508  
Intercept 281.46444263546846  
Prediction\_local [281.08616079]  
Right: 280.90103  
Intercept 281.49006071413413  
Prediction\_local [281.06195219]  
Right: 280.88864  
Intercept 281.59612044715595  
Prediction\_local [280.96991344]  
Right: 281.07922  
Intercept 281.56662234266645  
Prediction\_local [280.95305679]  
Right: 280.8469  
Intercept 281.1930393605639  
Prediction\_local [281.58860135]  
Right: 281.53128  
Intercept 281.23588533797397  
Prediction\_local [281.60548887]  
Right: 281.45517  
Intercept 281.44745761527525  
Prediction\_local [281.51724346]  
Right: 281.74725  
Intercept 281.39509372374556  
Prediction\_local [281.52964399]  
Right: 281.30637  
Intercept 281.604334493811  
Prediction\_local [280.91979664]  
Right: 280.86127  
Intercept 281.50008661805737  
Prediction\_local [281.12668038]  
Right: 281.2284  
Intercept 281.43248092168716  
Prediction\_local [281.1227762]  
Right: 280.59415  
Intercept 281.40367219095396  
Prediction\_local [281.14597035]  
Right: 281.00882  
Intercept 281.227640720643  
Prediction\_local [281.83942489]  
Right: 281.9652  
Intercept 281.4179485233462  
Prediction\_local [281.09319869]  
Right: 280.85754  
Intercept 281.4075317241019  
Prediction\_local [281.26927301]  
Right: 280.6664  
Intercept 281.30568754162965  
Prediction\_local [281.25614311]  
Right: 281.25085  
Intercept 281.2429580889007  
Prediction\_local [281.82183168]  
Right: 281.836  
Intercept 281.23085332983874  
Prediction\_local [281.58898607]  
Right: 281.60693  
Intercept 281.3700382590832  
Prediction\_local [281.56187653]  
Right: 281.8741  
Intercept 281.21053909393885  
Prediction\_local [281.76399013]  
Right: 281.86447  
Intercept 281.1804557950933  
Prediction\_local [281.86554951]  
Right: 281.90378  
Intercept 281.18709371044764  
Prediction\_local [281.6281181]  
Right: 281.26517

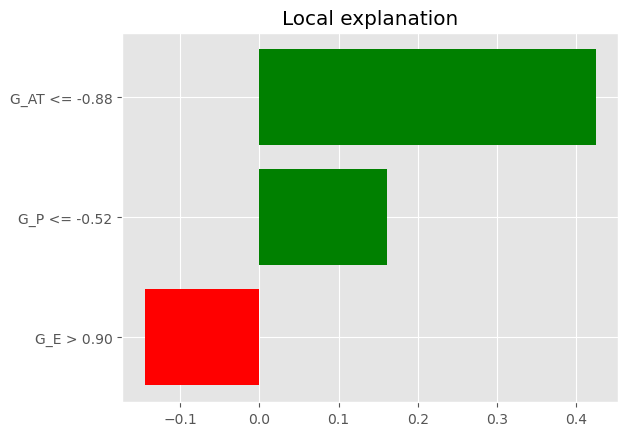
exp

<lime.explanation.Explanation at 0x25d96e9b940>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= -0.88', 0.42500866888050004),  
 ('G\_P <= -0.52', 0.16060047523445775),  
 ('G\_E > 0.90', -0.14458475175934904)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.42500866888050004),  
 (2, -0.16060047523445775),  
 (0, 0.14458475175934904)],  
 1: [(1, 0.42500866888050004),  
 (2, 0.16060047523445775),  
 (0, -0.14458475175934904)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.6281181]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.26517

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb2GR\_train.html')

#########################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, xgb2.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 281.309858706507  
Prediction\_local [281.7581397]  
Right: 281.6242  
Intercept 281.3660927926871  
Prediction\_local [281.5248277]  
Right: 281.64337  
Intercept 281.36635779214447  
Prediction\_local [281.05396308]  
Right: 280.98004  
Intercept 281.2100635899688  
Prediction\_local [281.76660684]  
Right: 281.85562  
Intercept 281.33832643542405  
Prediction\_local [281.32334997]  
Right: 281.33606  
Intercept 281.47495850918006  
Prediction\_local [281.41199726]  
Right: 281.23157  
Intercept 281.4351145557082  
Prediction\_local [281.59586965]  
Right: 281.2693  
Intercept 281.22733871584563  
Prediction\_local [281.49873263]  
Right: 281.37976  
Intercept 281.3619453142758  
Prediction\_local [281.38970535]  
Right: 281.3381  
Intercept 281.2851207460252  
Prediction\_local [281.56155198]  
Right: 281.49515  
Intercept 281.5332760509791  
Prediction\_local [281.01547794]  
Right: 281.04724  
Intercept 281.0390717622447  
Prediction\_local [281.80024711]  
Right: 281.8548  
Intercept 281.3671280082156  
Prediction\_local [281.63995011]  
Right: 281.46744  
Intercept 281.24492961336654  
Prediction\_local [281.77849243]  
Right: 282.01553  
Intercept 281.2845491502198  
Prediction\_local [281.31634529]  
Right: 281.28796  
Intercept 281.57059853975454  
Prediction\_local [281.0837003]  
Right: 280.8761  
Intercept 281.42026094493065  
Prediction\_local [281.75555263]  
Right: 281.8375  
Intercept 281.2847478334485  
Prediction\_local [281.92107543]  
Right: 281.83823  
Intercept 281.37061890506055  
Prediction\_local [281.1891929]  
Right: 281.46756  
Intercept 281.5492080472122  
Prediction\_local [281.16075613]  
Right: 281.01993  
Intercept 281.2437695543562  
Prediction\_local [281.8069406]  
Right: 282.0362  
Intercept 281.4848347968389  
Prediction\_local [281.35649264]  
Right: 281.1938  
Intercept 281.2325725948269  
Prediction\_local [281.5103381]  
Right: 281.46573  
Intercept 281.4536356102792  
Prediction\_local [281.47919353]  
Right: 281.49652  
Intercept 281.39147729702705  
Prediction\_local [281.91246257]  
Right: 282.09686  
Intercept 281.4945925628126  
Prediction\_local [280.9822333]  
Right: 281.0027  
Intercept 281.4414335923467  
Prediction\_local [281.66529186]  
Right: 281.4959

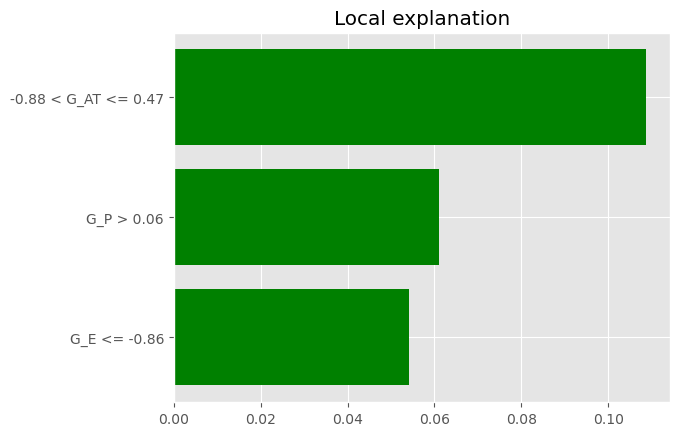
exp

<lime.explanation.Explanation at 0x25d95a7b0d0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('-0.88 < G\_AT <= 0.47', 0.10876945750233531),  
 ('G\_P > 0.06', 0.06094960712796684),  
 ('G\_E <= -0.86', 0.054139199370205636)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.10876945750233531),  
 (2, -0.06094960712796684),  
 (0, -0.054139199370205636)],  
 1: [(1, 0.10876945750233531),  
 (2, 0.06094960712796684),  
 (0, 0.054139199370205636)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [281.66529186]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 281.4959

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb2GR\_test.html')

#######################################################################################################

#######################################################################################################

#######################################################################################################

############ MODEL 3: RS Features (X\_R) and GT Target (LL\_G)###############

########### On the whole dataset  
model = XGBRegressor()  
xgb\_param\_grid = { 'colsample\_bytree' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'learning\_rate' : [0.001, 0.1, 0.20, 0.30],  
 'max\_depth' : [3, 6, 9, 12],  
 'n\_estimators' : [100, 500, 1000],  
 'subsample' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb3 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb3.fit(X\_R, y\_G)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'learning\_rate': [0.001, 0.1, 0.2, 0.3],  
 'max\_depth': [3, 6, 9, 12],  
 'n\_estimators': [100, 500, 1000],  
 'subsample': [0.1, 0.3, 0.5, 0.7, 0.9, 1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The best parameters are:', xgb3.best\_params\_)  
print('The best score is:', xgb3.best\_score\_)

The best parameters are: {'colsample\_bytree': 1, 'learning\_rate': 0.1, 'max\_depth': 9, 'n\_estimators': 100, 'subsample': 0.1}  
The best score is: -0.16240925377590898

#The best parameters are: {'colsample\_bytree': 1, 'learning\_rate': 0.1, 'max\_depth': 9, 'n\_estimators': 100, 'subsample': 0.1}  
#The best score is: -0.16240925377590898

# Evaluation of the performance of the model on the whole dataset:  
xgb\_y\_pred = xgb3.predict(X\_R)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_G, xgb\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_G, xgb\_y\_pred))  
print('R2 Score:', r2\_score(y\_G, xgb\_y\_pred))

Mean Absolute Error: 0.2258161078559045  
Mean Squared Error: 0.08503334725648314  
R2 Score: 0.7202921098395834

############## LIME ##############

###### Initializing the explainer using X\_R  
explainer = LimeTabularExplainer(X\_R.values, feature\_names = X\_R.columns, class\_names = ['LL\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the full XG datasset  
XR\_indx\_list = X\_R.index.tolist()  
XR\_dict = {}  
for n in XR\_indx\_list:  
 exp = explainer.explain\_instance(X\_R.loc[n].values, xgb3.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XR\_dict[n] = a

Intercept 279.62630110474066  
Prediction\_local [280.08178837]  
Right: 280.3851  
Intercept 279.55869357030485  
Prediction\_local [280.26244448]  
Right: 280.17007  
Intercept 279.72471577487346  
Prediction\_local [279.88454271]  
Right: 280.09653  
Intercept 279.7812330445167  
Prediction\_local [279.56606937]  
Right: 279.59143  
Intercept 279.89240811391954  
Prediction\_local [279.49361297]  
Right: 279.42172  
Intercept 279.86805780422753  
Prediction\_local [279.69186349]  
Right: 279.5525  
Intercept 279.9151744994382  
Prediction\_local [279.4558918]  
Right: 279.19016  
Intercept 279.9480011382666  
Prediction\_local [279.38046978]  
Right: 279.34647  
Intercept 279.86855374675037  
Prediction\_local [279.43959524]  
Right: 279.4986  
Intercept 279.58553736794454  
Prediction\_local [280.21644666]  
Right: 279.9891  
Intercept 279.65258021499096  
Prediction\_local [280.14969031]  
Right: 280.33658  
Intercept 279.64044049329243  
Prediction\_local [280.34902318]  
Right: 280.46896  
Intercept 279.8609567551501  
Prediction\_local [279.77919704]  
Right: 280.18088  
Intercept 279.623126703655  
Prediction\_local [280.23819405]  
Right: 280.0732  
Intercept 279.77474718605777  
Prediction\_local [279.84732514]  
Right: 279.85464  
Intercept 279.94310534593217  
Prediction\_local [279.31721321]  
Right: 279.37064  
Intercept 279.7908518041693  
Prediction\_local [279.64306868]  
Right: 279.51865  
Intercept 279.78170931690426  
Prediction\_local [279.72060122]  
Right: 279.51877  
Intercept 279.7898852075308  
Prediction\_local [279.58401014]  
Right: 279.2689  
Intercept 279.9058284416064  
Prediction\_local [279.59945297]  
Right: 279.36633  
Intercept 279.77590081177624  
Prediction\_local [279.69957881]  
Right: 279.46857  
Intercept 279.71573276075145  
Prediction\_local [279.68499111]  
Right: 279.62228  
Intercept 279.6088100061369  
Prediction\_local [280.26283692]  
Right: 280.41776  
Intercept 279.6189213044657  
Prediction\_local [280.34254234]  
Right: 280.50546  
Intercept 279.7200011673151  
Prediction\_local [279.93886259]  
Right: 280.00674  
Intercept 279.64529386728947  
Prediction\_local [280.23128195]  
Right: 280.04095  
Intercept 279.69138709952693  
Prediction\_local [279.76437054]  
Right: 279.82547  
Intercept 279.76342455204394  
Prediction\_local [279.7899147]  
Right: 279.83688  
Intercept 279.8789290372903  
Prediction\_local [279.44271528]  
Right: 279.0126  
Intercept 279.85014858944703  
Prediction\_local [279.60111166]  
Right: 279.55106  
Intercept 279.94580068664925  
Prediction\_local [279.47779732]  
Right: 279.38132  
Intercept 279.8684356506577  
Prediction\_local [279.45947806]  
Right: 279.4807  
Intercept 279.9152924988548  
Prediction\_local [279.45823022]  
Right: 279.5096  
Intercept 279.74653844585964  
Prediction\_local [279.66490943]  
Right: 279.85568  
Intercept 279.64718402732177  
Prediction\_local [280.35670385]  
Right: 280.44333  
Intercept 279.60692446109914  
Prediction\_local [280.24162778]  
Right: 280.1623  
Intercept 279.76847597685526  
Prediction\_local [279.92700616]  
Right: 280.0722  
Intercept 279.63604565508405  
Prediction\_local [280.1741346]  
Right: 280.0565  
Intercept 279.75588320741565  
Prediction\_local [279.71397413]  
Right: 279.73315  
Intercept 279.7483457778014  
Prediction\_local [279.7118747]  
Right: 279.54285  
Intercept 279.79605942051523  
Prediction\_local [279.58552437]  
Right: 279.53488  
Intercept 279.81275558143034  
Prediction\_local [279.39994877]  
Right: 279.16974  
Intercept 280.0090069433758  
Prediction\_local [279.27475768]  
Right: 279.18918  
Intercept 279.93062601698693  
Prediction\_local [279.50857863]  
Right: 279.2988  
Intercept 279.9436989625907  
Prediction\_local [279.32378653]  
Right: 279.37393  
Intercept 279.8568308905003  
Prediction\_local [279.56089444]  
Right: 279.64032  
Intercept 279.6002382459386  
Prediction\_local [280.21977783]  
Right: 280.43674  
Intercept 279.5679651919832  
Prediction\_local [280.32124962]  
Right: 280.36685  
Intercept 279.66275889331246  
Prediction\_local [279.91565854]  
Right: 280.12537  
Intercept 279.63068746314445  
Prediction\_local [280.30967064]  
Right: 279.97437  
Intercept 279.7390552340485  
Prediction\_local [279.91045932]  
Right: 279.79422  
Intercept 279.88479695201556  
Prediction\_local [279.40770561]  
Right: 279.4793  
Intercept 279.840617800017  
Prediction\_local [279.51872556]  
Right: 279.36542  
Intercept 279.95514642055065  
Prediction\_local [279.42765989]  
Right: 279.3219  
Intercept 279.9270737049601  
Prediction\_local [279.29310643]  
Right: 279.3207  
Intercept 279.956783036373  
Prediction\_local [279.30583849]  
Right: 279.01874  
Intercept 279.8582166332428  
Prediction\_local [279.55864449]  
Right: 279.43295  
Intercept 279.92035664730685  
Prediction\_local [279.56615103]  
Right: 279.84598  
Intercept 279.6474594339808  
Prediction\_local [280.32273882]  
Right: 280.44443  
Intercept 279.6621462135819  
Prediction\_local [280.30000388]  
Right: 280.511  
Intercept 279.76078292300394  
Prediction\_local [279.87952441]  
Right: 280.0551  
Intercept 279.6795597463777  
Prediction\_local [280.19334156]  
Right: 279.98068  
Intercept 279.8544736631413  
Prediction\_local [279.44215122]  
Right: 279.54764  
Intercept 279.78483978670124  
Prediction\_local [279.71099605]  
Right: 279.63477  
Intercept 279.92617968304245  
Prediction\_local [279.40324171]  
Right: 279.44492  
Intercept 279.9169540433536  
Prediction\_local [279.39440092]  
Right: 279.3016  
Intercept 279.932802031711  
Prediction\_local [279.39818206]  
Right: 279.29184  
Intercept 279.8927714859421  
Prediction\_local [279.6060561]  
Right: 279.47052  
Intercept 279.9158960347314  
Prediction\_local [279.44250394]  
Right: 279.45575  
Intercept 279.90490789145144  
Prediction\_local [279.66659345]  
Right: 279.92117  
Intercept 279.6357468628429  
Prediction\_local [280.28141665]  
Right: 280.47083  
Intercept 279.6921550657976  
Prediction\_local [280.21555992]  
Right: 280.39905  
Intercept 279.72603274825747  
Prediction\_local [279.92296629]  
Right: 280.0836  
Intercept 279.6764146060369  
Prediction\_local [280.19622011]  
Right: 280.1474  
Intercept 279.787994206894  
Prediction\_local [279.85986663]  
Right: 279.7707  
Intercept 279.94358875087306  
Prediction\_local [279.31550674]  
Right: 279.63116  
Intercept 279.96285235343674  
Prediction\_local [279.34355676]  
Right: 279.38873  
Intercept 279.9692250627799  
Prediction\_local [279.29429954]  
Right: 279.41522  
Intercept 280.0006406248731  
Prediction\_local [279.37801085]  
Right: 279.2659  
Intercept 279.8995138532727  
Prediction\_local [279.47825583]  
Right: 279.2841  
Intercept 279.92422591925816  
Prediction\_local [279.43830348]  
Right: 279.72025  
Intercept 279.8972893519233  
Prediction\_local [279.35494901]  
Right: 279.42578  
Intercept 279.6191743843862  
Prediction\_local [280.45083647]  
Right: 280.46155  
Intercept 279.654223853133  
Prediction\_local [280.29878921]  
Right: 280.35822  
Intercept 279.77370453615805  
Prediction\_local [279.9160536]  
Right: 280.0956  
Intercept 279.63663471104314  
Prediction\_local [280.2622567]  
Right: 280.22238  
Intercept 279.873308748751  
Prediction\_local [279.81677615]  
Right: 279.884  
Intercept 279.8287423530175  
Prediction\_local [279.69223629]  
Right: 279.54657  
Intercept 279.8460229066833  
Prediction\_local [279.61359681]  
Right: 279.72586  
Intercept 279.8665809401376  
Prediction\_local [279.55888228]  
Right: 279.5124  
Intercept 280.0066796990545  
Prediction\_local [279.39166997]  
Right: 279.28174  
Intercept 279.94345471637104  
Prediction\_local [279.50358366]  
Right: 279.3878  
Intercept 279.88619743774893  
Prediction\_local [279.49385206]  
Right: 279.62814  
Intercept 279.8422178668337  
Prediction\_local [279.59900632]  
Right: 279.91534  
Intercept 279.5675713792309  
Prediction\_local [280.33569825]  
Right: 280.5247  
Intercept 279.59671756262304  
Prediction\_local [280.29107572]  
Right: 280.5139  
Intercept 279.73549489473345  
Prediction\_local [280.00138954]  
Right: 280.20294  
Intercept 279.6019988641539  
Prediction\_local [280.3375909]  
Right: 280.23325  
Intercept 279.5587805774523  
Prediction\_local [280.38332485]  
Right: 280.5143  
Intercept 279.8702196971301  
Prediction\_local [279.7625834]  
Right: 279.72638  
Intercept 279.7872700101866  
Prediction\_local [279.85996102]  
Right: 279.6197  
Intercept 279.763264399072  
Prediction\_local [279.9216304]  
Right: 279.58524  
Intercept 279.89085713569415  
Prediction\_local [279.44707015]  
Right: 279.4453  
Intercept 279.875095521829  
Prediction\_local [279.54030295]  
Right: 279.4189  
Intercept 279.8636220391499  
Prediction\_local [279.42928795]  
Right: 279.736  
Intercept 279.6593569024263  
Prediction\_local [280.09488819]  
Right: 280.41327  
Intercept 279.6420855485791  
Prediction\_local [280.21436231]  
Right: 280.31186  
Intercept 279.7422014329297  
Prediction\_local [279.95416139]  
Right: 280.46942

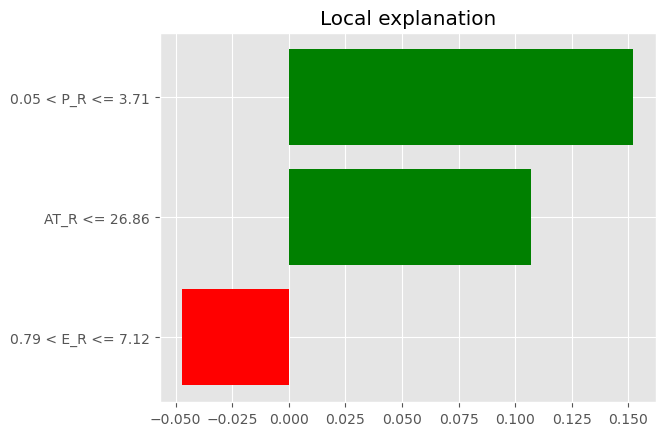
exp

<lime.explanation.Explanation at 0x18b60d326d0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('0.05 < P\_R <= 3.71', 0.15214878943922877),  
 ('AT\_R <= 26.86', 0.10690710110741188),  
 ('0.79 < E\_R <= 7.12', -0.047095929213541096)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.15214878943922877),  
 (1, -0.10690710110741188),  
 (0, 0.047095929213541096)],  
 1: [(2, 0.15214878943922877),  
 (1, 0.10690710110741188),  
 (0, -0.047095929213541096)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.95416139]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.46942

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb3RG\_Whole.html')

#################################################################################################

#The best parameters are: {'colsample\_bytree': 1, 'learning\_rate': 0.1, 'max\_depth': 9, 'n\_estimators': 100, 'subsample': 0.1}  
#The best score is: -0.16240925377590898

########### Fitting the model on the training dataset using the best parameters  
model = XGBRegressor()  
xgb\_param\_grid = {'colsample\_bytree' : [1],  
 'learning\_rate' : [0.1],  
 'max\_depth' : [9],  
 'n\_estimators' : [100],  
 'subsample' : [0.1],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb3 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb3.fit(X\_R\_train, y\_G\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 max\_cat\_to\_onehot=None, max\_delta\_step=None,  
 max\_depth=None, max\_leaves=None,  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [1], 'learning\_rate': [0.1],  
 'max\_depth': [9], 'n\_estimators': [100],  
 'subsample': [0.1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The best score is:', xgb3.best\_score\_)

The best score is: -0.17750998317263933

#######################################################

# Initialization  
model = XGBRegressor(colsample\_bytree = 1, learning\_rate = 0.1, max\_depth = 9, n\_estimators = 100, subsample = 0.1 )

# Fitting the model  
xgb3 = model.fit(X\_R\_train, y\_G\_train)

# Evaluation of the performance of the model on the training dataset:  
xgb\_y\_predtr = xgb3.predict(X\_R\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, xgb\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, xgb\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, xgb\_y\_predtr))

The training MAE is: 0.1947311288339106  
The training MSE is: 0.07446623442042387  
The training R2 Score is: 0.7557652469045069

# Evaluation of the performance of the model on the testing dataset:  
xgb\_y\_predts = xgb3.predict(X\_R\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, xgb\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, xgb\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, xgb\_y\_predts))

The testing MAE is: 0.3310518391927109  
The testing MSE is: 0.16865471084289196  
The testing R2 Score is: 0.3849928159316147

#####################Cross-validation ##############################

############# On the training dataset  
   
score\_xgb3tr = cross\_val\_score(xgb3, X\_R\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_xgb3tr)))

The mean training CV score is: 0.16674135228389134

############# On the testing dataset   
score\_xgb3ts = cross\_val\_score(xgb3, X\_R\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_xgb3ts)))

The mean testing CV score is: 0.19290545048231963

############## LIME ##############

###### Initializing the explainer using the training dataset  
explainer = LimeTabularExplainer(X\_R\_train.values, feature\_names = X\_R\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_R\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_train.loc[n].values, xgb3.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 279.54497023145495  
Prediction\_local [279.75173391]  
Right: 279.77972  
Intercept 279.600890474461  
Prediction\_local [279.65452389]  
Right: 279.6801  
Intercept 279.460205681557  
Prediction\_local [280.24078171]  
Right: 280.45047  
Intercept 279.85451637537597  
Prediction\_local [279.16415468]  
Right: 279.04782  
Intercept 279.75770590061535  
Prediction\_local [279.31224617]  
Right: 279.37842  
Intercept 279.4884199943038  
Prediction\_local [280.06073741]  
Right: 280.23026  
Intercept 279.86307658160064  
Prediction\_local [279.13651472]  
Right: 279.13287  
Intercept 279.7964905699927  
Prediction\_local [279.47054684]  
Right: 279.4321  
Intercept 279.76725114212763  
Prediction\_local [279.2622191]  
Right: 279.26962  
Intercept 279.89308367727193  
Prediction\_local [279.05901052]  
Right: 279.1102  
Intercept 279.8278216005104  
Prediction\_local [279.22289829]  
Right: 279.5059  
Intercept 279.73452930130236  
Prediction\_local [279.59725544]  
Right: 279.5929  
Intercept 279.70030045350353  
Prediction\_local [279.95405543]  
Right: 280.02194  
Intercept 279.8036459458279  
Prediction\_local [279.35919626]  
Right: 279.22675  
Intercept 279.4546253252836  
Prediction\_local [280.22882411]  
Right: 280.4166  
Intercept 279.81223265419067  
Prediction\_local [279.22883135]  
Right: 278.99854  
Intercept 279.5609010850991  
Prediction\_local [280.09092441]  
Right: 280.2773  
Intercept 279.8079508896066  
Prediction\_local [279.34294816]  
Right: 279.6041  
Intercept 279.7109977728071  
Prediction\_local [279.62243768]  
Right: 279.55634  
Intercept 279.7534839113394  
Prediction\_local [279.27462767]  
Right: 279.03625  
Intercept 279.7873072899603  
Prediction\_local [279.25834295]  
Right: 279.0887  
Intercept 279.5854739049756  
Prediction\_local [280.11721757]  
Right: 280.40793  
Intercept 279.50737913674686  
Prediction\_local [279.91013702]  
Right: 279.7511  
Intercept 279.81052512292445  
Prediction\_local [279.23415723]  
Right: 279.43497  
Intercept 279.5737590609181  
Prediction\_local [279.83772347]  
Right: 279.83456  
Intercept 279.4800120559402  
Prediction\_local [280.11040999]  
Right: 280.26132  
Intercept 279.7657358522939  
Prediction\_local [279.54982173]  
Right: 279.43796  
Intercept 279.5261149195937  
Prediction\_local [280.01188159]  
Right: 280.26797  
Intercept 279.8454189499584  
Prediction\_local [279.05663358]  
Right: 278.94925  
Intercept 279.43315705223887  
Prediction\_local [280.20373641]  
Right: 280.0497  
Intercept 279.7027654132074  
Prediction\_local [279.32720177]  
Right: 279.5763  
Intercept 279.55919926793075  
Prediction\_local [280.01292475]  
Right: 280.2195  
Intercept 279.5227094970131  
Prediction\_local [280.01456023]  
Right: 280.28506  
Intercept 279.48371488635405  
Prediction\_local [280.15600562]  
Right: 280.2773  
Intercept 279.48218952856803  
Prediction\_local [280.15122927]  
Right: 280.395  
Intercept 279.852142237514  
Prediction\_local [279.00245379]  
Right: 279.38318  
Intercept 279.58333749428806  
Prediction\_local [280.00842812]  
Right: 279.9644  
Intercept 279.5344035022056  
Prediction\_local [280.20074718]  
Right: 280.17255  
Intercept 279.7611785670188  
Prediction\_local [279.53632856]  
Right: 279.54352  
Intercept 279.6681561412118  
Prediction\_local [279.52195962]  
Right: 279.8852  
Intercept 279.52685140607025  
Prediction\_local [279.99830814]  
Right: 280.1327  
Intercept 279.6969144757667  
Prediction\_local [279.29426814]  
Right: 279.44934  
Intercept 279.6101907491974  
Prediction\_local [279.87209242]  
Right: 279.9529  
Intercept 279.58940211799427  
Prediction\_local [280.11674732]  
Right: 280.2389  
Intercept 279.8066883453327  
Prediction\_local [279.16091738]  
Right: 279.16266  
Intercept 279.8602285022827  
Prediction\_local [279.20544543]  
Right: 279.01443  
Intercept 279.59152978918286  
Prediction\_local [279.95376956]  
Right: 279.87735  
Intercept 279.52198994609404  
Prediction\_local [280.08873429]  
Right: 280.1029  
Intercept 279.49727311459714  
Prediction\_local [280.07877407]  
Right: 279.9774  
Intercept 279.7662267165719  
Prediction\_local [279.28368628]  
Right: 279.09106  
Intercept 279.7438606762855  
Prediction\_local [279.61760941]  
Right: 279.35715  
Intercept 279.4138133289612  
Prediction\_local [280.17655997]  
Right: 280.40018  
Intercept 279.50599971586024  
Prediction\_local [280.04472614]  
Right: 280.3182  
Intercept 279.6866548795529  
Prediction\_local [279.59275866]  
Right: 279.36356  
Intercept 279.8097160586756  
Prediction\_local [279.38860608]  
Right: 279.50403  
Intercept 279.543337287341  
Prediction\_local [280.06391293]  
Right: 280.3101  
Intercept 279.7120428575594  
Prediction\_local [279.69723204]  
Right: 279.6472  
Intercept 279.8269608212986  
Prediction\_local [279.55642299]  
Right: 279.46286  
Intercept 279.6867515670654  
Prediction\_local [279.35633224]  
Right: 279.06982  
Intercept 279.66238807977726  
Prediction\_local [279.52872608]  
Right: 279.468  
Intercept 279.72891388306067  
Prediction\_local [279.44980098]  
Right: 279.16122  
Intercept 279.77698740561414  
Prediction\_local [279.33974977]  
Right: 279.41315  
Intercept 279.6898908084466  
Prediction\_local [279.47236783]  
Right: 279.5355  
Intercept 279.51024897144174  
Prediction\_local [280.16729194]  
Right: 280.30057  
Intercept 279.50599647760475  
Prediction\_local [280.18984686]  
Right: 280.03082  
Intercept 279.73634384800226  
Prediction\_local [279.36893961]  
Right: 279.40448  
Intercept 279.56327860181875  
Prediction\_local [280.07170815]  
Right: 280.00766  
Intercept 279.7796460334281  
Prediction\_local [279.25513128]  
Right: 279.09943  
Intercept 279.73456851581415  
Prediction\_local [279.33092684]  
Right: 279.06934  
Intercept 279.87174371022894  
Prediction\_local [279.21248695]  
Right: 278.98657  
Intercept 279.6873298669929  
Prediction\_local [279.6322306]  
Right: 279.54355  
Intercept 279.4749365094184  
Prediction\_local [280.20121274]  
Right: 280.29886  
Intercept 279.81713045901563  
Prediction\_local [279.65397713]  
Right: 279.30295  
Intercept 279.8681693757182  
Prediction\_local [279.09899639]  
Right: 279.11218  
Intercept 279.6243308568223  
Prediction\_local [279.87838748]  
Right: 279.9411  
Intercept 279.54260363681686  
Prediction\_local [280.11351455]  
Right: 280.43137  
Intercept 279.6064439613203  
Prediction\_local [279.97178491]  
Right: 280.1696  
Intercept 279.7149554968047  
Prediction\_local [279.56833619]  
Right: 279.59363  
Intercept 279.6727663694659  
Prediction\_local [279.66815894]  
Right: 280.03812  
Intercept 279.5504490778319  
Prediction\_local [280.13559248]  
Right: 280.34512  
Intercept 279.5382163964055  
Prediction\_local [279.99342584]  
Right: 279.89896

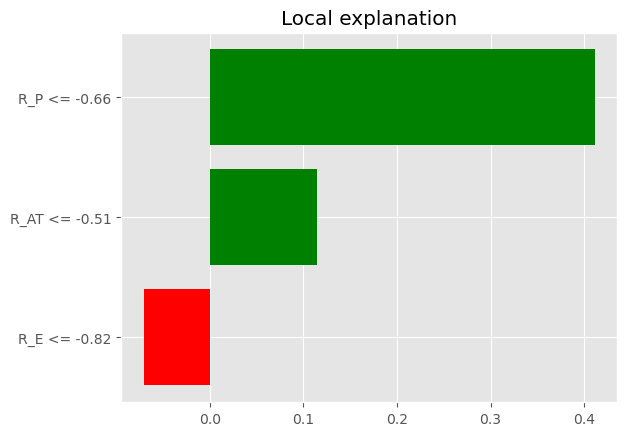
exp

<lime.explanation.Explanation at 0x25d97582fa0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P <= -0.66', 0.4114317581895451),  
 ('R\_AT <= -0.51', 0.11403792379738682),  
 ('R\_E <= -0.82', -0.0702602375248344)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, -0.4114317581895451),  
 (1, -0.11403792379738682),  
 (0, 0.0702602375248344)],  
 1: [(2, 0.4114317581895451),  
 (1, 0.11403792379738682),  
 (0, -0.0702602375248344)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.99342584]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.89896

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb3RG\_train.html')

######################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_R\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_R\_test.loc[n].values, xgb3.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.4419132008658  
Prediction\_local [280.17208236]  
Right: 279.9875  
Intercept 279.4339403972309  
Prediction\_local [280.06882729]  
Right: 280.34262  
Intercept 279.7539447889435  
Prediction\_local [279.5564505]  
Right: 279.57364  
Intercept 279.6700690735555  
Prediction\_local [279.61772131]  
Right: 279.6116  
Intercept 279.7470262224058  
Prediction\_local [279.51474233]  
Right: 279.71817  
Intercept 279.7136554832502  
Prediction\_local [279.61624196]  
Right: 279.5716  
Intercept 279.7487298951524  
Prediction\_local [279.61529023]  
Right: 279.80563  
Intercept 279.67756899852134  
Prediction\_local [279.85424178]  
Right: 280.19656  
Intercept 279.80146885039835  
Prediction\_local [279.34966601]  
Right: 279.27826  
Intercept 279.5672151752264  
Prediction\_local [280.06130867]  
Right: 280.25894  
Intercept 279.7029299276155  
Prediction\_local [279.43530429]  
Right: 279.3422  
Intercept 279.7391068235691  
Prediction\_local [279.62247412]  
Right: 279.68182  
Intercept 279.54208445255506  
Prediction\_local [280.025266]  
Right: 280.00958  
Intercept 279.4939025952066  
Prediction\_local [279.99163068]  
Right: 279.95547  
Intercept 279.59565385026735  
Prediction\_local [279.70588087]  
Right: 279.7251  
Intercept 279.6421406922234  
Prediction\_local [279.53643324]  
Right: 279.40817  
Intercept 279.70329882414796  
Prediction\_local [279.53003973]  
Right: 279.4445  
Intercept 279.7523316962555  
Prediction\_local [279.80031638]  
Right: 279.70694  
Intercept 279.8204674285323  
Prediction\_local [279.3090926]  
Right: 279.0825  
Intercept 279.8140425735724  
Prediction\_local [279.35773533]  
Right: 279.0825  
Intercept 279.5211747404327  
Prediction\_local [280.07390928]  
Right: 280.15375  
Intercept 279.9248194103837  
Prediction\_local [279.10231412]  
Right: 279.13297  
Intercept 279.6659295430316  
Prediction\_local [280.15176786]  
Right: 280.10043  
Intercept 279.571496586315  
Prediction\_local [279.96613262]  
Right: 280.05875  
Intercept 279.58413802650523  
Prediction\_local [280.18457711]  
Right: 280.3619  
Intercept 279.6010869779875  
Prediction\_local [279.4057455]  
Right: 279.5823  
Intercept 279.7257668363399  
Prediction\_local [279.32914953]  
Right: 279.34943

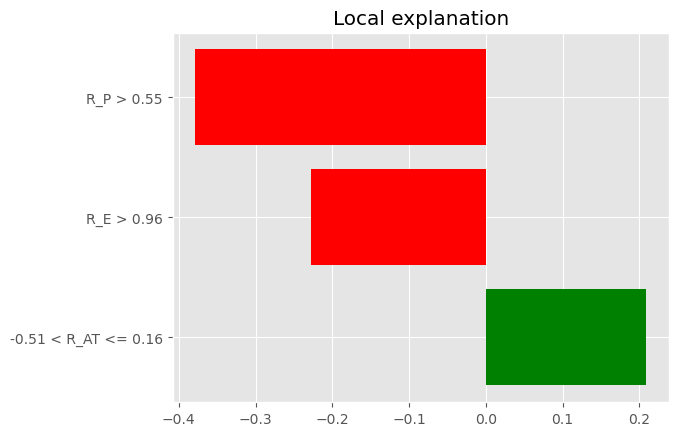
exp

<lime.explanation.Explanation at 0x25d97549160>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('R\_P > 0.55', -0.3784413016917756),  
 ('R\_E > 0.96', -0.22732374863594793),  
 ('-0.51 < R\_AT <= 0.16', 0.2091477452606184)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(2, 0.3784413016917756),  
 (0, 0.22732374863594793),  
 (1, -0.2091477452606184)],  
 1: [(2, -0.3784413016917756),  
 (0, -0.22732374863594793),  
 (1, 0.2091477452606184)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.32914953]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.34943

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb3RG\_test.html')

#############################################################################################

#############################################################################################

#############################################################################################

############ MODEL 4: GT Features (X\_G) and GT Target (LL\_G)###############

########### On the whole dataset  
model = XGBRegressor()  
xgb\_param\_grid = { 'colsample\_bytree' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'learning\_rate' : [0.001, 0.1, 0.20, 0.30],  
 'max\_depth' : [3, 6, 9, 12],  
 'n\_estimators' : [100, 500, 1000],  
 'subsample' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb4 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb4.fit(X\_G, y\_G)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [0.1, 0.3, 0.5, 0.7, 0.9, 1],  
 'learning\_rate': [0.001, 0.1, 0.2, 0.3],  
 'max\_depth': [3, 6, 9, 12],  
 'n\_estimators': [100, 500, 1000],  
 'subsample': [0.1, 0.3, 0.5, 0.7, 0.9, 1]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best parameters and the best score  
print('The best parameters are:', xgb4.best\_params\_)  
print('The best score is:', xgb4.best\_score\_)

The best parameters are: {'colsample\_bytree': 1, 'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100, 'subsample': 0.5}  
The best score is: -0.0980750933574402

#The best parameters are: {'colsample\_bytree': 1, 'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100, 'subsample': 0.5}  
#The best score is: -0.0980750933574402

# Evaluation of the performance of the model on the whole dataset:  
xgb\_y\_pred = xgb4.predict(X\_G)  
print('Mean Absolute Error:', mean\_absolute\_error(y\_G, xgb\_y\_pred))  
print('Mean Squared Error:', mean\_squared\_error(y\_G, xgb\_y\_pred))  
print('R2 Score:', r2\_score(y\_G, xgb\_y\_pred))

Mean Absolute Error: 0.10347363507306302  
Mean Squared Error: 0.01779896365264833  
R2 Score: 0.94145225689743

############### LIME #################

###### Initializing the explainer using the whole X\_G  
explainer = LimeTabularExplainer(X\_G.values, feature\_names = X\_G.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the whole XG datasset  
XG\_indx\_list = X\_G.index.tolist()  
XG\_dict = {}  
for n in XG\_indx\_list:  
 exp = explainer.explain\_instance(X\_G.loc[n].values, xgb4.predict, num\_features = 3, num\_samples = 108)  
 a = exp.as\_list()  
 XG\_dict[n] = a

Intercept 279.54567937273094  
Prediction\_local [280.44141022]  
Right: 280.38248  
Intercept 279.5543726969118  
Prediction\_local [280.35152695]  
Right: 280.19714  
Intercept 279.8114416830127  
Prediction\_local [279.57621843]  
Right: 279.79782  
Intercept 279.7510167948943  
Prediction\_local [279.62613331]  
Right: 279.5467  
Intercept 279.7920288679155  
Prediction\_local [279.49667454]  
Right: 279.27118  
Intercept 280.0608179713305  
Prediction\_local [279.29195672]  
Right: 279.2368  
Intercept 279.9863353055571  
Prediction\_local [279.29631193]  
Right: 279.24893  
Intercept 279.9462387680626  
Prediction\_local [279.55430874]  
Right: 279.0512  
Intercept 279.8990701644042  
Prediction\_local [279.96164471]  
Right: 279.72153  
Intercept 279.54371880308184  
Prediction\_local [280.24117755]  
Right: 279.99234  
Intercept 279.65888886501597  
Prediction\_local [280.40557314]  
Right: 280.49176  
Intercept 279.5119076559024  
Prediction\_local [280.46824937]  
Right: 280.41122  
Intercept 279.44373865527615  
Prediction\_local [280.47185607]  
Right: 280.17496  
Intercept 279.59137188665403  
Prediction\_local [280.25425485]  
Right: 280.1258  
Intercept 279.91062852002824  
Prediction\_local [279.6366573]  
Right: 279.826  
Intercept 279.8958255337072  
Prediction\_local [279.53622515]  
Right: 279.55328  
Intercept 279.84891382200743  
Prediction\_local [279.51781981]  
Right: 279.41406  
Intercept 279.7543821284325  
Prediction\_local [279.51906624]  
Right: 279.24695  
Intercept 280.10628469740556  
Prediction\_local [279.35356921]  
Right: 278.98703  
Intercept 279.97320401086097  
Prediction\_local [279.43565121]  
Right: 279.0499  
Intercept 280.0281976641372  
Prediction\_local [279.75238015]  
Right: 279.46686  
Intercept 279.44358091857947  
Prediction\_local [280.22059837]  
Right: 279.9983  
Intercept 279.5779683257835  
Prediction\_local [280.21317526]  
Right: 280.50772  
Intercept 279.64267382733846  
Prediction\_local [280.43343703]  
Right: 280.40732  
Intercept 279.56134418089465  
Prediction\_local [280.33724123]  
Right: 280.20468  
Intercept 279.4990260281297  
Prediction\_local [280.37947058]  
Right: 280.04852  
Intercept 279.8011719175794  
Prediction\_local [279.38081724]  
Right: 279.8149  
Intercept 279.74206213971405  
Prediction\_local [279.5964895]  
Right: 279.45627  
Intercept 279.8438585170476  
Prediction\_local [279.38642575]  
Right: 278.702  
Intercept 280.1165641823579  
Prediction\_local [279.0513294]  
Right: 279.14474  
Intercept 279.8713601405327  
Prediction\_local [279.85482819]  
Right: 279.0314  
Intercept 279.89567727339863  
Prediction\_local [279.87313047]  
Right: 278.99838  
Intercept 279.88789425447146  
Prediction\_local [279.57265844]  
Right: 279.19196  
Intercept 279.55846967133886  
Prediction\_local [280.15739696]  
Right: 279.76224  
Intercept 279.66871473518324  
Prediction\_local [280.30074255]  
Right: 280.09555  
Intercept 279.5339682429336  
Prediction\_local [280.30728077]  
Right: 279.9827  
Intercept 279.5971057399714  
Prediction\_local [280.24490591]  
Right: 279.86017  
Intercept 279.59184522775024  
Prediction\_local [280.19095756]  
Right: 279.7255  
Intercept 279.6992805072539  
Prediction\_local [279.66236365]  
Right: 279.65152  
Intercept 279.88368884805163  
Prediction\_local [279.48651594]  
Right: 279.59384  
Intercept 280.0769166577604  
Prediction\_local [279.47716779]  
Right: 279.59985  
Intercept 280.142956811689  
Prediction\_local [279.188808]  
Right: 278.8771  
Intercept 280.1385297210291  
Prediction\_local [279.35557001]  
Right: 278.8697  
Intercept 279.84763201645444  
Prediction\_local [279.67465335]  
Right: 279.02338  
Intercept 279.825318527819  
Prediction\_local [280.18483937]  
Right: 279.70435  
Intercept 279.70405357355037  
Prediction\_local [279.70115125]  
Right: 280.06912  
Intercept 279.4663753721877  
Prediction\_local [280.14183684]  
Right: 280.32983  
Intercept 279.48622132820293  
Prediction\_local [280.54732754]  
Right: 280.40628  
Intercept 279.4723923253246  
Prediction\_local [280.3778125]  
Right: 280.0663  
Intercept 279.6112776137675  
Prediction\_local [280.25021556]  
Right: 279.96213  
Intercept 279.503943691331  
Prediction\_local [280.08433649]  
Right: 279.77182  
Intercept 280.0109520321194  
Prediction\_local [279.42303182]  
Right: 279.46906  
Intercept 279.8871791109226  
Prediction\_local [279.56658971]  
Right: 279.35587  
Intercept 279.92285504030514  
Prediction\_local [279.29315192]  
Right: 279.06155  
Intercept 280.0183668872983  
Prediction\_local [279.33523119]  
Right: 278.88647  
Intercept 280.039748194762  
Prediction\_local [279.43926563]  
Right: 278.9937  
Intercept 279.9439091853509  
Prediction\_local [279.85506192]  
Right: 279.48538  
Intercept 279.6698554461733  
Prediction\_local [279.82105145]  
Right: 280.0181  
Intercept 279.63128927509257  
Prediction\_local [280.24267129]  
Right: 280.34995  
Intercept 279.60685745278784  
Prediction\_local [280.53941238]  
Right: 280.46027  
Intercept 279.5475765466085  
Prediction\_local [280.23510741]  
Right: 280.20773  
Intercept 279.61630063126654  
Prediction\_local [280.048545]  
Right: 280.0075  
Intercept 279.7870789074633  
Prediction\_local [279.7213511]  
Right: 279.73224  
Intercept 279.8598200619966  
Prediction\_local [279.5636865]  
Right: 279.6227  
Intercept 279.84660091544  
Prediction\_local [279.44831701]  
Right: 279.1907  
Intercept 280.037207155729  
Prediction\_local [279.32816552]  
Right: 279.1319  
Intercept 279.9803207577967  
Prediction\_local [279.35523806]  
Right: 279.10553  
Intercept 279.8264249845833  
Prediction\_local [279.76184653]  
Right: 279.62686  
Intercept 279.85812305629105  
Prediction\_local [279.92462964]  
Right: 279.98718  
Intercept 279.91768004473886  
Prediction\_local [280.00626021]  
Right: 280.3317  
Intercept 279.4772456448661  
Prediction\_local [280.4024876]  
Right: 280.56158  
Intercept 279.447692278262  
Prediction\_local [280.44649081]  
Right: 280.4432  
Intercept 279.56636672007363  
Prediction\_local [280.36384753]  
Right: 280.12512  
Intercept 279.6014938488183  
Prediction\_local [280.23729841]  
Right: 280.003  
Intercept 279.8583165589106  
Prediction\_local [279.65100255]  
Right: 279.76004  
Intercept 279.82902326197626  
Prediction\_local [279.40393502]  
Right: 279.54922  
Intercept 279.73886033279837  
Prediction\_local [279.58987154]  
Right: 279.34167  
Intercept 280.15963726704445  
Prediction\_local [279.03115422]  
Right: 279.10858  
Intercept 280.05354892103713  
Prediction\_local [279.43043476]  
Right: 279.17834  
Intercept 279.90113403554415  
Prediction\_local [279.37128553]  
Right: 279.40176  
Intercept 279.95372564864385  
Prediction\_local [279.87112034]  
Right: 280.19812  
Intercept 279.9463572093046  
Prediction\_local [279.7214066]  
Right: 280.04044  
Intercept 279.5836008245536  
Prediction\_local [280.19795196]  
Right: 280.57144  
Intercept 279.5715874130248  
Prediction\_local [280.39455308]  
Right: 280.4432  
Intercept 279.4535515582149  
Prediction\_local [280.39792314]  
Right: 280.5217  
Intercept 279.54802772738583  
Prediction\_local [280.37503078]  
Right: 280.34427  
Intercept 279.67200220678  
Prediction\_local [279.68100402]  
Right: 279.98654  
Intercept 279.7836918823377  
Prediction\_local [279.48682464]  
Right: 279.79172  
Intercept 279.91712480352675  
Prediction\_local [279.44414012]  
Right: 279.36954  
Intercept 280.02995617256687  
Prediction\_local [279.29764209]  
Right: 279.46283  
Intercept 280.1040207369892  
Prediction\_local [279.49780232]  
Right: 279.29547  
Intercept 279.94394172253016  
Prediction\_local [279.48985799]  
Right: 279.528  
Intercept 279.9990778485324  
Prediction\_local [279.73614545]  
Right: 280.08932  
Intercept 279.5483795748754  
Prediction\_local [280.33072168]  
Right: 280.24557  
Intercept 279.7725242459405  
Prediction\_local [280.10477531]  
Right: 280.54358  
Intercept 279.46268181223036  
Prediction\_local [280.55811089]  
Right: 280.61383  
Intercept 279.45332466770657  
Prediction\_local [280.43068701]  
Right: 280.48166  
Intercept 279.63378685420776  
Prediction\_local [280.23940591]  
Right: 280.4376  
Intercept 279.52317638416883  
Prediction\_local [280.00953675]  
Right: 279.93497  
Intercept 279.98933948483534  
Prediction\_local [279.52263072]  
Right: 279.58368  
Intercept 279.8803823817234  
Prediction\_local [279.59423828]  
Right: 279.61655  
Intercept 279.90069338488655  
Prediction\_local [279.46065658]  
Right: 279.3658  
Intercept 280.04802203306457  
Prediction\_local [279.45555036]  
Right: 279.15213  
Intercept 280.00666857065346  
Prediction\_local [279.50034815]  
Right: 279.17218  
Intercept 279.8306827040968  
Prediction\_local [279.74873442]  
Right: 279.3563  
Intercept 279.6643642688831  
Prediction\_local [280.21850104]  
Right: 280.03864  
Intercept 279.8415831419924  
Prediction\_local [280.20531833]  
Right: 280.33267  
Intercept 279.5357741652321  
Prediction\_local [280.40100781]  
Right: 280.4815

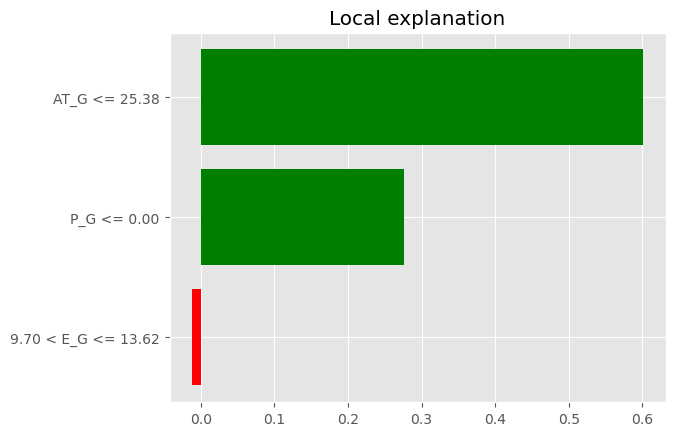
exp

<lime.explanation.Explanation at 0x18b5d9a97f0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('AT\_G <= 25.38', 0.6009517316528525),  
 ('P\_G <= 0.00', 0.27620556400705637),  
 ('9.70 < E\_G <= 13.62', -0.011923651327605733)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.6009517316528525),  
 (2, -0.27620556400705637),  
 (0, 0.011923651327605733)],  
 1: [(1, 0.6009517316528525),  
 (2, 0.27620556400705637),  
 (0, -0.011923651327605733)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.40100781]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 280.4815

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb4GG\_Whole.html')

##############################################################################

#The best parameters are: {'colsample\_bytree': 1, 'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100, 'subsample': 0.5}  
#The best score is: -0.0980750933574402

########### Fitting the model on the training dataset using the best parameters  
model = XGBRegressor()  
xgb\_param\_grid = {'colsample\_bytree' : [1],  
 'learning\_rate' : [0.1],  
 'max\_depth' : [3],  
 'n\_estimators' : [100],  
 'subsample' : [0.5],  
 }  
xgb\_cv = KFold(n\_splits = 10, random\_state = 1, shuffle = True)  
xgb4 = GridSearchCV(model, xgb\_param\_grid, cv = xgb\_cv, scoring = 'neg\_mean\_squared\_error', n\_jobs = -1)  
xgb4.fit(X\_G\_train, y\_G\_train)

GridSearchCV(cv=KFold(n\_splits=10, random\_state=1, shuffle=True),  
 estimator=XGBRegressor(base\_score=None, booster=None,  
 callbacks=None, colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None, device=None,  
 early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None,  
 feature\_types=None, gamma=None,  
 grow\_policy=None, importance\_type=None,  
 inte...  
 max\_cat\_to\_onehot=None, max\_delta\_step=None,  
 max\_depth=None, max\_leaves=None,  
 min\_child\_weight=None, missing=nan,  
 monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=None,  
 n\_jobs=None, num\_parallel\_tree=None,  
 random\_state=None, ...),  
 n\_jobs=-1,  
 param\_grid={'colsample\_bytree': [1], 'learning\_rate': [0.1],  
 'max\_depth': [3], 'n\_estimators': [100],  
 'subsample': [0.5]},  
 scoring='neg\_mean\_squared\_error')

# Printing the best score  
print('The best score is:', xgb4.best\_score\_)

The best score is: -0.11711069695470352

################################################

# Initialization  
model = XGBRegressor(colsample\_bytree = 1, learning\_rate = 0.1, max\_depth = 3, n\_estimators = 100, subsample = 0.5 )

# Fitting the model  
xgb4 = model.fit(X\_G\_train, y\_G\_train)

# Evaluation of the performance of the model on the training dataset:  
xgb\_y\_predtr = xgb4.predict(X\_G\_train)  
print('The training MAE is:', mean\_absolute\_error(y\_G\_train, xgb\_y\_predtr))  
print('The training MSE is:', mean\_squared\_error(y\_G\_train, xgb\_y\_predtr))  
print('The training R2 Score is:', r2\_score(y\_G\_train, xgb\_y\_predtr))

The training MAE is: 0.08908991307388474  
The training MSE is: 0.013321892324476886  
The training R2 Score is: 0.9563067864521834

# Evaluation of the performance of the model on the testing dataset:  
xgb\_y\_predts = xgb4.predict(X\_G\_test)  
print('The testing MAE is:', mean\_absolute\_error(y\_G\_test, xgb\_y\_predts))  
print('The testing MSE is:', mean\_squared\_error(y\_G\_test, xgb\_y\_predts))  
print('The testing R2 Score is:', r2\_score(y\_G\_test, xgb\_y\_predts))

The testing MAE is: 0.2723366970486154  
The testing MSE is: 0.11851546747331698  
The testing R2 Score is: 0.5678278800809144

#####################Cross-validation ##############################

############# On the training dataset  
score\_xgb4tr = cross\_val\_score(xgb4, X\_G\_train, y\_G\_train, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean training score  
print('The mean training CV score is:', absolute(np.mean(score\_xgb4tr)))

The mean training CV score is: 0.10901744886278561

############# On the testing dataset   
score\_xgb4ts = cross\_val\_score(xgb4, X\_G\_test, y\_G\_test, scoring = 'neg\_mean\_squared\_error', cv = 10)

# The mean testing score  
print('The mean testing CV score is:', absolute(np.mean(score\_xgb4ts)))

The mean testing CV score is: 0.1314127776285276

############### LIME #################

###### Initializing the explainer using the training dataset  
explainer = LimeTabularExplainer(X\_G\_train.values, feature\_names = X\_G\_train.columns, class\_names = ['y\_G'],  
verbose = True, mode = 'regression')

# Explanation of the prediction for the training datasset  
train\_indx\_list = X\_G\_train.index.tolist()  
train\_dict = {}  
for n in train\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_train.loc[n].values, xgb4.predict, num\_features = 3, num\_samples = 81)  
 a = exp.as\_list()  
 train\_dict[n] = a

Intercept 279.62363352449313  
Prediction\_local [279.523277]  
Right: 279.80237  
Intercept 279.6942524766271  
Prediction\_local [279.3816421]  
Right: 279.6081  
Intercept 279.5309319769222  
Prediction\_local [280.17622926]  
Right: 280.42953  
Intercept 279.955887895837  
Prediction\_local [279.0899885]  
Right: 278.84528  
Intercept 279.6954867096504  
Prediction\_local [279.48637773]  
Right: 279.0344  
Intercept 279.4228868334715  
Prediction\_local [280.23174039]  
Right: 280.4501  
Intercept 279.75782389155194  
Prediction\_local [279.59341191]  
Right: 279.29636  
Intercept 279.72992097001475  
Prediction\_local [279.20262719]  
Right: 279.26663  
Intercept 279.8274214537812  
Prediction\_local [279.13994226]  
Right: 279.20685  
Intercept 279.8612279512443  
Prediction\_local [279.37513924]  
Right: 279.1977  
Intercept 279.5603581110085  
Prediction\_local [279.89480941]  
Right: 280.14587  
Intercept 279.6167187058033  
Prediction\_local [279.47865155]  
Right: 279.63742  
Intercept 279.6754734440392  
Prediction\_local [279.61326395]  
Right: 280.00107  
Intercept 279.80694390101496  
Prediction\_local [279.38975378]  
Right: 279.16083  
Intercept 279.3913277216081  
Prediction\_local [280.32493689]  
Right: 280.51038  
Intercept 279.8898372955994  
Prediction\_local [278.98659527]  
Right: 278.99725  
Intercept 279.50344044057186  
Prediction\_local [280.08042874]  
Right: 280.07422  
Intercept 279.7992648126589  
Prediction\_local [279.65846619]  
Right: 280.0545  
Intercept 279.7321491537727  
Prediction\_local [279.33073684]  
Right: 279.16034  
Intercept 279.7083515719978  
Prediction\_local [279.58434732]  
Right: 279.04272  
Intercept 279.8170612441244  
Prediction\_local [279.19461929]  
Right: 279.12704  
Intercept 279.3275415509482  
Prediction\_local [280.30004686]  
Right: 280.48087  
Intercept 279.5925760919249  
Prediction\_local [279.59556861]  
Right: 279.80402  
Intercept 279.70100868274113  
Prediction\_local [279.45538026]  
Right: 279.44934  
Intercept 279.6824641240731  
Prediction\_local [279.54318827]  
Right: 279.8101  
Intercept 279.52688353791353  
Prediction\_local [280.06964627]  
Right: 280.4881  
Intercept 279.84844495605404  
Prediction\_local [279.43396139]  
Right: 279.2991  
Intercept 279.5211812669085  
Prediction\_local [280.08901575]  
Right: 280.01556  
Intercept 279.8981800441441  
Prediction\_local [279.14006497]  
Right: 278.84775  
Intercept 279.4729344526391  
Prediction\_local [280.07877015]  
Right: 280.0392  
Intercept 279.8139276921713  
Prediction\_local [279.45097708]  
Right: 279.56787  
Intercept 279.41181197090685  
Prediction\_local [280.20807356]  
Right: 280.4715  
Intercept 279.4905292200698  
Prediction\_local [280.15186794]  
Right: 279.8112  
Intercept 279.49816216021316  
Prediction\_local [279.83578544]  
Right: 280.51526  
Intercept 279.43780144545696  
Prediction\_local [280.28745869]  
Right: 280.36642  
Intercept 279.7968440814418  
Prediction\_local [279.51947971]  
Right: 279.43118  
Intercept 279.44627806496953  
Prediction\_local [280.06264215]  
Right: 279.9998  
Intercept 279.56703449501157  
Prediction\_local [279.86343155]  
Right: 279.98923  
Intercept 279.68531570554865  
Prediction\_local [279.43708843]  
Right: 279.62686  
Intercept 279.634969397589  
Prediction\_local [280.27999261]  
Right: 280.38788  
Intercept 279.54011351534206  
Prediction\_local [279.97975363]  
Right: 280.33475  
Intercept 279.5930153548386  
Prediction\_local [279.85411221]  
Right: 279.61758  
Intercept 279.5642024878887  
Prediction\_local [280.036907]  
Right: 280.143  
Intercept 279.48942465410715  
Prediction\_local [279.80220104]  
Right: 280.4015  
Intercept 279.7694044361171  
Prediction\_local [279.1962617]  
Right: 278.64902  
Intercept 279.8430497955024  
Prediction\_local [279.15649916]  
Right: 279.00146  
Intercept 279.7098250698643  
Prediction\_local [279.80334406]  
Right: 279.8531  
Intercept 279.4546306068558  
Prediction\_local [279.88989722]  
Right: 279.63574  
Intercept 279.54142788486683  
Prediction\_local [280.05996347]  
Right: 280.06842  
Intercept 279.81450260574303  
Prediction\_local [279.34903795]  
Right: 279.1044  
Intercept 279.7243794925808  
Prediction\_local [279.41961549]  
Right: 279.5395  
Intercept 279.4776757709391  
Prediction\_local [280.24485341]  
Right: 280.42645  
Intercept 279.4449904230981  
Prediction\_local [280.16940959]  
Right: 280.49942  
Intercept 279.7322707251509  
Prediction\_local [279.37924293]  
Right: 279.2214  
Intercept 279.6919683714976  
Prediction\_local [279.76535013]  
Right: 279.99615  
Intercept 279.5711072831787  
Prediction\_local [279.79094701]  
Right: 280.0591  
Intercept 279.76375000995387  
Prediction\_local [279.49210711]  
Right: 279.49304  
Intercept 279.6784656192741  
Prediction\_local [279.56260649]  
Right: 279.83438  
Intercept 279.7243102325895  
Prediction\_local [279.53779088]  
Right: 279.07446  
Intercept 279.76301685141425  
Prediction\_local [279.48775366]  
Right: 279.50778  
Intercept 279.7514168449266  
Prediction\_local [279.25482872]  
Right: 279.42105  
Intercept 279.79995433977757  
Prediction\_local [279.57237283]  
Right: 279.5835  
Intercept 279.9562978975078  
Prediction\_local [279.16210225]  
Right: 278.93018  
Intercept 279.6020841686436  
Prediction\_local [280.08263846]  
Right: 279.94705  
Intercept 279.5092266595843  
Prediction\_local [279.82273256]  
Right: 280.00882  
Intercept 279.7351452389726  
Prediction\_local [279.69026816]  
Right: 279.47946  
Intercept 279.5672142239491  
Prediction\_local [280.0150705]  
Right: 279.9638  
Intercept 279.85013206300306  
Prediction\_local [279.1734021]  
Right: 279.1999  
Intercept 279.866347949808  
Prediction\_local [279.15718302]  
Right: 279.0502  
Intercept 279.7461947596767  
Prediction\_local [279.32118408]  
Right: 278.97424  
Intercept 279.78721335269546  
Prediction\_local [279.42555018]  
Right: 279.61612  
Intercept 279.5531829751701  
Prediction\_local [280.06941284]  
Right: 280.243  
Intercept 279.66431353437645  
Prediction\_local [279.43543948]  
Right: 279.24252  
Intercept 279.79261691808284  
Prediction\_local [279.59565303]  
Right: 279.07718  
Intercept 279.51032640163436  
Prediction\_local [279.60790405]  
Right: 279.83304  
Intercept 279.43977762752945  
Prediction\_local [280.23525626]  
Right: 280.35645  
Intercept 279.4746901301722  
Prediction\_local [280.22408365]  
Right: 280.16528  
Intercept 279.5305042884036  
Prediction\_local [280.01812505]  
Right: 279.9788  
Intercept 279.52347058712724  
Prediction\_local [280.00235756]  
Right: 280.0965  
Intercept 279.336925704811  
Prediction\_local [280.26009531]  
Right: 280.3969  
Intercept 279.5466405862491  
Prediction\_local [280.01719817]  
Right: 279.7929

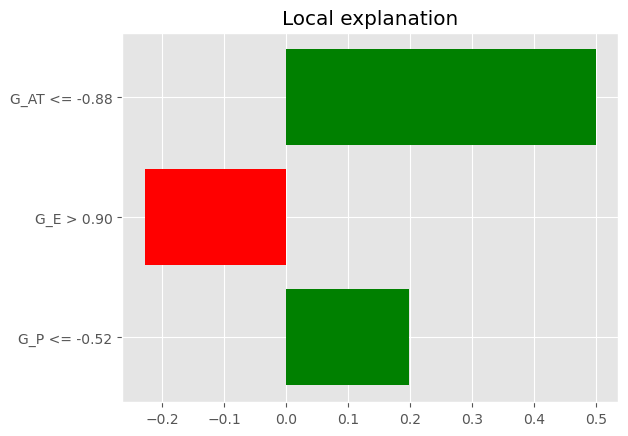
exp

<lime.explanation.Explanation at 0x25d9755cd30>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_AT <= -0.88', 0.5000147794557339),  
 ('G\_E > 0.90', -0.22781704609204032),  
 ('G\_P <= -0.52', 0.19835985048400248)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(1, -0.5000147794557339),  
 (0, 0.22781704609204032),  
 (2, -0.19835985048400248)],  
 1: [(1, 0.5000147794557339),  
 (0, -0.22781704609204032),  
 (2, 0.19835985048400248)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [280.01719817]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.7929

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb4GG\_train.html')

################################################################################

# Explanation of the prediction for the testing datasset  
test\_indx\_list = X\_G\_test.index.tolist()  
test\_dict = {}  
for n in test\_indx\_list:  
 exp = explainer.explain\_instance(X\_G\_test.loc[n].values, xgb4.predict, num\_features = 3, num\_samples = 27)  
 a = exp.as\_list()  
 test\_dict[n] = a

Intercept 279.5665703723138  
Prediction\_local [280.201987]  
Right: 280.15836  
Intercept 279.49464406732073  
Prediction\_local [280.20039953]  
Right: 280.17743  
Intercept 279.59280390088827  
Prediction\_local [279.51067812]  
Right: 279.53754  
Intercept 279.65334273443153  
Prediction\_local [279.8176067]  
Right: 279.891  
Intercept 279.59704119997537  
Prediction\_local [279.81587805]  
Right: 279.6318  
Intercept 279.67696564303475  
Prediction\_local [279.41193225]  
Right: 279.36832  
Intercept 279.7759329314975  
Prediction\_local [279.5494867]  
Right: 279.25735  
Intercept 279.508087535049  
Prediction\_local [280.02805748]  
Right: 279.9835  
Intercept 279.7311486526626  
Prediction\_local [279.57586932]  
Right: 279.33017  
Intercept 279.4489461828118  
Prediction\_local [280.05070436]  
Right: 280.12314  
Intercept 279.7088996333428  
Prediction\_local [279.53867417]  
Right: 279.62518  
Intercept 279.4158818798048  
Prediction\_local [280.0022114]  
Right: 279.82596  
Intercept 279.4773654368281  
Prediction\_local [280.0901103]  
Right: 280.07562  
Intercept 279.75986769519164  
Prediction\_local [280.16353482]  
Right: 280.46585  
Intercept 279.7377525480829  
Prediction\_local [279.57395666]  
Right: 279.80527  
Intercept 279.93037180136713  
Prediction\_local [279.2734043]  
Right: 279.0549  
Intercept 279.7647485802852  
Prediction\_local [279.72538794]  
Right: 279.74005  
Intercept 279.7200759171984  
Prediction\_local [279.83897387]  
Right: 280.0253  
Intercept 279.89824959065703  
Prediction\_local [279.37105101]  
Right: 279.1718  
Intercept 279.81071864261946  
Prediction\_local [279.2173284]  
Right: 279.00928  
Intercept 279.5714307295328  
Prediction\_local [280.14868604]  
Right: 280.45877  
Intercept 279.84604326364524  
Prediction\_local [279.1321548]  
Right: 279.12613  
Intercept 279.40362729377347  
Prediction\_local [280.04252615]  
Right: 280.04697  
Intercept 279.6908501442748  
Prediction\_local [280.04688032]  
Right: 280.2163  
Intercept 279.51725744811495  
Prediction\_local [280.28177808]  
Right: 280.50104  
Intercept 279.6724296760002  
Prediction\_local [279.3963912]  
Right: 279.5395  
Intercept 279.7225488829699  
Prediction\_local [279.91226376]  
Right: 279.55948

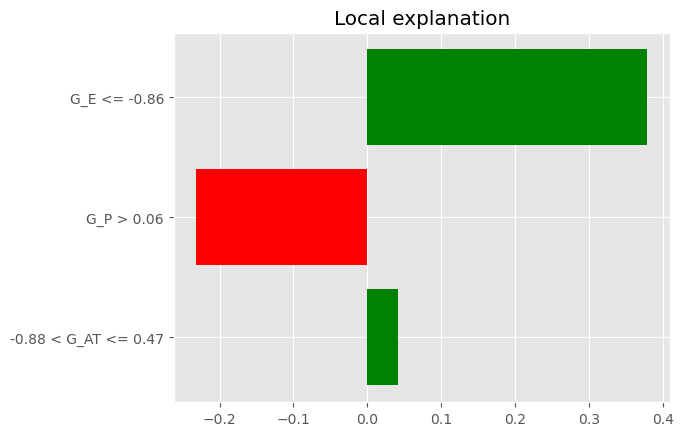
exp

<lime.explanation.Explanation at 0x25d9755c2b0>

##### Showing the explanable table  
  
exp.show\_in\_notebook(show\_table = True)

<IPython.core.display.HTML object>

###### Visualization of feature importances using as\_pyplot\_figure()  
import matplotlib.pyplot as plt  
with plt.style.context('ggplot'):  
 exp.as\_pyplot\_figure()



## Retrieving feature importances as a list using as\_list()  
  
exp.as\_list()

[('G\_E <= -0.86', 0.3791413850512398),  
 ('G\_P > 0.06', -0.23139355875182085),  
 ('-0.88 < G\_AT <= 0.47', 0.04196705385288238)]

## Retrieving feature importances as dictionary using as\_map()  
exp.as\_map()

{0: [(0, -0.3791413850512398),  
 (2, 0.23139355875182085),  
 (1, -0.04196705385288238)],  
 1: [(0, 0.3791413850512398),  
 (2, -0.23139355875182085),  
 (1, 0.04196705385288238)]}

# Retrieving feature importances as htnl using as\_html() function  
from IPython.display import HTML  
html\_data = exp.as\_html()  
HTML(data = html\_data)

<IPython.core.display.HTML object>

# Retrieving Average Local and Global Predicted Values

print('Éxplanation Local Prediction:', exp.local\_pred)

Éxplanation Local Prediction: [279.91226376]

print('Éxplanation Gobal Prediction:', exp.predicted\_value)

Éxplanation Gobal Prediction: 279.55948

# Saving feature importances to Html file using save\_to\_file() function  
  
exp.save\_to\_file('XAILL\_gb4GG\_test.html')

##################### END ####################################