Ouput from regressor_models_features.py

$ python regressor_models_features.py

C:\Users\juliette.courgibet\AppData\Roaming\Python\Python37\site-packages\sklearn\model_selection_split.py:296: FutureWarning: Setting a random_state has no effect since shuffle is False. This will raise an error in 0.24. You should leave random_state to its default (None), or set shuffle=True.

  FutureWarning

'''

Getting features for train dataset

0 : KNeighborsRegressor(algorithm='auto', leaf_size=10, metric='minkowski',

metric_params=None, n_jobs=None, n_neighbors=13, p=1,

weights='distance')

iteration 1 :: mse= 0.2204491391069131 RMSE= 0.46952011576386493

iteration 2 :: mse= 0.23514204270029265 RMSE= 0.4849144694688875

iteration 3 :: mse= 0.23773206946819955 RMSE= 0.48757775735589043

iteration 4 :: mse= 0.22201203024373475 RMSE= 0.47118152578781647

iteration 5 :: mse= 0.228227156873214 RMSE= 0.47773126009631606

iteration 6 :: mse= 0.2299639931972767 RMSE= 0.4795456111750755

iteration 7 :: mse= 0.2321199542038599 RMSE= 0.4817882877404347

iteration 8 :: mse= 0.23139048504417425 RMSE= 0.48103064875761736

iteration 9 :: mse= 0.21644766857436076 RMSE= 0.46523936696539425

iteration 10 :: mse= 0.22903196396456807 RMSE= 0.4785728408137763

Modele terminé

1 : LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=True)

iteration 1 :: mse= 0.22302053410547162 RMSE= 0.47225049931733437

iteration 2 :: mse= 0.24008414959542249 RMSE= 0.4899838258508361

iteration 3 :: mse= 0.24658739845585226 RMSE= 0.49657567243659073

iteration 4 :: mse= 0.22403764517881763 RMSE= 0.4733261509559953

iteration 5 :: mse= 0.23143463863325478 RMSE= 0.48107654134581823

iteration 6 :: mse= 0.23484399872703332 RMSE= 0.4846070560021112

iteration 7 :: mse= 0.2353567437361118 RMSE= 0.48513579927285494

iteration 8 :: mse= 0.23081550104882273 RMSE= 0.4804326186353532

iteration 9 :: mse= 0.22336409511362215 RMSE= 0.4726141080348979

iteration 10 :: mse= 0.23276198231372258 RMSE= 0.48245412456908543

Modele terminé

2 : BaggingRegressor(base_estimator=None, bootstrap=True, bootstrap_features=False,
max_features=0.6, max_samples=0.8, n_estimators=500,
n_jobs=None, oob_score=False, random_state=None, verbose=0,
warm_start=False)

iteration 1 :: mse= 0.20974278505453534 RMSE= 0.45797683899356234

iteration 2 :: mse= 0.22519467540207516 RMSE= 0.47454681054883846

iteration 3 :: mse= 0.22547021911684592 RMSE= 0.4748370448025785

iteration 4 :: mse= 0.21099920593673482 RMSE= 0.45934649877487344

iteration 5 :: mse= 0.2167956659918069 RMSE= 0.465613215009848

iteration 6 :: mse= 0.21882941260422295 RMSE= 0.46779206128815715

iteration 7 :: mse= 0.22042898450676093 RMSE= 0.4694986522949357

iteration 8 :: mse= 0.21815444534861278 RMSE= 0.46707006471043805

iteration 9 :: mse= 0.20769674837117408 RMSE= 0.4557375871827713

iteration 10 :: mse= 0.21792892949088086 RMSE= 0.46682858683983874

Modele terminé

3 : DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=7,
max_features='sqrt', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=None, splitter='best')

iteration 1 :: mse= 0.22857335847846796 RMSE= 0.4780934620745906

iteration 2 :: mse= 0.2512306668216202 RMSE= 0.5012291559971549

iteration 3 :: mse= 0.254120053687862 RMSE= 0.5041032172956864

iteration 4 :: mse= 0.23209579779451933 RMSE= 0.48176321756078405

iteration 5 :: mse= 0.23672201536447876 RMSE= 0.4865408671062265

iteration 6 :: mse= 0.24374963112927445 RMSE= 0.493710067883241

iteration 7 :: mse= 0.24194996078783762 RMSE= 0.4918840928387882

iteration 8 :: mse= 0.23585841482072412 RMSE= 0.485652565957109

iteration 9 :: mse= 0.22988094599429534 RMSE= 0.47945901388366385

iteration 10 :: mse= 0.23995444783006678 RMSE= 0.4898514548616415

Modele terminé

4 : RandomForestRegressor(bootstrap=False, ccp_alpha=0.0, criterion='mse',
max_depth=7, max_features=0.4, max_leaf_nodes=None,
max_samples=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=1,
min_samples_split=2, min_weight_fraction_leaf=0.0,
n_estimators=100, n_jobs=None, oob_score=False,
random_state=None, verbose=0, warm_start=False)

iteration 1 :: mse= 0.22107062100331276 RMSE= 0.47018147666971394

iteration 2 :: mse= 0.24217170190996432 RMSE= 0.49210944098844955

iteration 3 :: mse= 0.24268478472727056 RMSE= 0.49263047482598005

iteration 4 :: mse= 0.22402214883592328 RMSE= 0.47330978104822985

iteration 5 :: mse= 0.22942279896465118 RMSE= 0.47898100063014104

iteration 6 :: mse= 0.23571791184517796 RMSE= 0.4855078906106243

iteration 7 :: mse= 0.23278079212340158 RMSE= 0.4824736180594765

iteration 8 :: mse= 0.22847826064471347 RMSE= 0.4779939964525846

iteration 9 :: mse= 0.22173263569102553 RMSE= 0.47088494952697896

iteration 10 :: mse= 0.23374087626387827 RMSE= 0.4834675545099984

Modele terminé

5 : MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
beta_2=0.999, early_stopping=False, epsilon=1e-08,
hidden_layer_sizes=(100,), learning_rate='constant',
learning_rate_init=0.001, max_fun=15000, max_iter=200,
momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
power_t=0.5, random_state=None, shuffle=True, solver='adam',
tol=0.0001, validation_fraction=0.1, verbose=False,
warm_start=False)

iteration 1 :: mse= 0.2229275447928019 RMSE= 0.4721520356758

iteration 2 :: mse= 0.2415768354579025 RMSE= 0.49150466473666604

iteration 3 :: mse= 0.2504751010347655 RMSE= 0.5004748755279984

iteration 4 :: mse= 0.22761038666814778 RMSE= 0.47708530334537425

iteration 5 :: mse= 0.22971533002896036 RMSE= 0.4792862714797497

iteration 6 :: mse= 0.23928544719116096 RMSE= 0.48916811751294764

iteration 7 :: mse= 0.236820770014658 RMSE= 0.48664234301451614

iteration 8 :: mse= 0.2437631779823181 RMSE= 0.4937237871343836

iteration 9 :: mse= 0.22611280860698024 RMSE= 0.4755132055022029

iteration 10 :: mse= 0.2328776559462071 RMSE= 0.4825739901260812

Modele terminé

6 : AdaBoostRegressor(base_estimator=None, learning_rate=1.0, loss='linear',
n_estimators=50, random_state=None)

iteration 1 :: mse= 0.23750049363251072 RMSE= 0.4873402236964549

iteration 2 :: mse= 0.24383235595365146 RMSE= 0.4937938395258202

iteration 3 :: mse= 0.24709085676514672 RMSE= 0.49708234404889773

iteration 4 :: mse= 0.24711287443964947 RMSE= 0.49710449046417743

iteration 5 :: mse= 0.24836620662250566 RMSE= 0.49836352858380967

iteration 6 :: mse= 0.2411722851848958 RMSE= 0.4910929496387581

iteration 7 :: mse= 0.2534546990291632 RMSE= 0.5034428458416736

iteration 8 :: mse= 0.2513958533522446 RMSE= 0.5013939103661358

iteration 9 :: mse= 0.23859593562352316 RMSE= 0.4884628293161345

iteration 10 :: mse= 0.2446031419321586 RMSE= 0.494573697169753

Modele terminé

7 : ExtraTreesRegressor(bootstrap=False, ccp_alpha=0.0, criterion='mse',
max_depth=20, max_features=None, max_leaf_nodes=None,
max_samples=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=1,
min_samples_split=2, min_weight_fraction_leaf=0.0,
n_estimators=100, n_jobs=None, oob_score=False,
random_state=None, verbose=0, warm_start=False)

iteration 1 :: mse= 0.21114225919345525 RMSE= 0.45950218627712236

iteration 2 :: mse= 0.22562265538464474 RMSE= 0.47499753197742484

iteration 3 :: mse= 0.22606705816341513 RMSE= 0.47546509668262205

iteration 4 :: mse= 0.2115438899122189 RMSE= 0.4599390067304782

iteration 5 :: mse= 0.2189818713657 RMSE= 0.46795498861076373

iteration 6 :: mse= 0.2210131747582088 RMSE= 0.4701203832617863

iteration 7 :: mse= 0.22210921392302058 RMSE= 0.47128464214635785

iteration 8 :: mse= 0.21761117451111994 RMSE= 0.46648812901414755

iteration 9 :: mse= 0.20958253600338872 RMSE= 0.4578018523372188

iteration 10 :: mse= 0.21928995592564007 RMSE= 0.4682840547420338

Modele terminé

8 : XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
colsample_bynode=1, colsample_bytree=1.0, gamma=0.1,
importance_type='gain', learning_rate=0.1, max_delta_step=0,
max_depth=7, min_child_weight=2, missing=None, n_estimators=100,
n_jobs=1, nthread=None, objective='reg:squarederror',
random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
seed=None, silent=None, subsample=1, verbosity=1)

iteration 1 :: mse= 0.21300287929266978 RMSE= 0.46152234972173317

iteration 2 :: mse= 0.22854284098004432 RMSE= 0.47806154518016225

iteration 3 :: mse= 0.22761779952052255 RMSE= 0.47709307217829366

iteration 4 :: mse= 0.21489925227475634 RMSE= 0.46357227297882725

iteration 5 :: mse= 0.22246582475615387 RMSE= 0.4716628295256622

iteration 6 :: mse= 0.2245216626055698 RMSE= 0.4738371688729893

iteration 7 :: mse= 0.22396893416502214 RMSE= 0.4732535622317302

iteration 8 :: mse= 0.22177268633217886 RMSE= 0.47092747459898626

iteration 9 :: mse= 0.21214656359987877 RMSE= 0.46059370772935965

iteration 10 :: mse= 0.2222476859180326 RMSE= 0.47143152834535007

Modele terminé

Getting features for test dataset

0 : KNeighborsRegressor(algorithm='auto', leaf_size=10, metric='minkowski',

metric_params=None, n_jobs=None, n_neighbors=13, p=1,
weights='distance')

Modele terminé

1 : LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=True)

Modele terminé

2 : BaggingRegressor(base_estimator=None, bootstrap=True, bootstrap_features=False,
max_features=0.6, max_samples=0.8, n_estimators=500,
n_jobs=None, oob_score=False, random_state=None, verbose=0,
warm_start=False)

Modele terminé

3 : DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=7,
max_features='sqrt', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=None, splitter='best')

Modele terminé

4 : RandomForestRegressor(bootstrap=False, ccp_alpha=0.0, criterion='mse',
max_depth=7, max_features=0.4, max_leaf_nodes=None,
max_samples=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=1,
min_samples_split=2, min_weight_fraction_leaf=0.0,
n_estimators=100, n_jobs=None, oob_score=False,
random_state=None, verbose=0, warm_start=False)

Modele terminé

5 : MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
beta_2=0.999, early_stopping=False, epsilon=1e-08,
hidden_layer_sizes=(100,), learning_rate='constant',
learning_rate_init=0.001, max_fun=15000, max_iter=200,
momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
power_t=0.5, random_state=None, shuffle=True, solver='adam',

tol=0.0001, validation_fraction=0.1, verbose=False,

warm_start=False)

Modele terminé

6 : AdaBoostRegressor(base_estimator=None, learning_rate=1.0, loss='linear',

n_estimators=50, random_state=None)

Modele terminé

7 : ExtraTreesRegressor(bootstrap=False, ccp_alpha=0.0, criterion='mse',

max_depth=20, max_features=None, max_leaf_nodes=None,

max_samples=None, min_impurity_decrease=0.0,

min_impurity_split=None, min_samples_leaf=1,

min_samples_split=2, min_weight_fraction_leaf=0.0,

n_estimators=100, n_jobs=None, oob_score=False,

random_state=None, verbose=0, warm_start=False)

Modele terminé

8 : XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,

colsample_bynode=1, colsample_bytree=1.0, gamma=0.1,

importance_type='gain', learning_rate=0.1, max_delta_step=0,

max_depth=7, min_child_weight=2, missing=None, n_estimators=100,

n_jobs=1, nthread=None, objective='reg:squarederror',

random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,

seed=None, silent=None, subsample=1, verbosity=1)

Modele terminé