In [189]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_absolute_percentage_error

from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression
```

In [190]:
```
df = pd.read_csv('wc-wo-outliers.csv')
```

In [191]:
```
df.head()
```

Out[191]:

	goals_z	xg_z	crosses_z	boxtouches_z	passes_z	progpasses_z	takeons_z	progruns_z
0	0.423077	0.146923	-0.136154	-0.030000	0.429231	0.037692	0.244615	-0.220000
1	0.479231	0.609231	0.227692	0.450769	0.770769	0.042308	0.337692	0.927692
2	0.877692	0.773846	0.428462	0.659231	0.754615	0.335385	0.023077	0.638462
3	0.245385	0.097692	0.549231	0.490000	0.090769	0.071538	-0.473077	-0.150769
4	0.337692	0.270000	0.292308	0.281538	0.065385	-0.142308	-0.076923	0.430000

In [192]:
```
df.shape
```

Out[192]:
```
(200, 17)
```

In [193]:
```
y = df['results']
X = df.drop(columns=['results'])

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=1)
```

In [194]:
```
regr = RandomForestRegressor(max_depth=20, n_estimators=1000, min_samples_split=2,
                             max_features=15, random_state=1)
regr.fit(X_train, y_train)
```

Out[194]:
```
RandomForestRegressor(max_depth=20, max_features=15, n_estimators=1000,
                      random_state=1)
```

In [195]:
```
y_pred = regr.predict(X_test)
```

In [196]:
```
print('DECISION FOREST REGRESSION')
print('r2 score: '+str(r2_score(y_test, y_pred)))
print('RMSE : '+str(np.sqrt(mean_squared_error(y_test, y_pred))))
print('MSE: '+str(mean_squared_error(y_test, y_pred)))
print('MAE: '+str(mean_absolute_error(y_test, y_pred)))
print('MAPE: '+str(mean_absolute_percentage_error(y_test, y_pred)))


print('-----------------')

y_train_pred = regr.predict(X_train)

print('r2 score: '+str(r2_score(y_train, y_train_pred)))
print('RMSE : '+str(np.sqrt(mean_squared_error(y_train, y_train_pred))))
print('MSE: '+str(mean_squared_error(y_train, y_train_pred)))
print('MAE: '+str(mean_absolute_error(y_train, y_train_pred)))
print('MAPE: '+str(mean_absolute_percentage_error(y_train, y_train_pred
)))
```

```
DECISION FOREST REGRESSION
r2 score: 0.5827604094022283
RMSE : 5.5224874015247788
MSE: 30.497867099999997
MAE: 4.62885
MAPE: 0.5495634426707304
-----------------
r2 score: 0.9086346713314737
RMSE : 2.7318704581293747
MSE: 7.463116199999999
MAE: 2.2874624999999997
MAPE: 0.3876610403279976
```

In [197]:
```
# mape .37432196450623323
# mape @ (178, 17) = .38923381998443746
# mape @ (189, 17) = .3718819197955769
# z < 2.3 for (189, 17) = (165, 17) = .3452481429512346
#mape @ (194, 17) with z < 2.7 == .366891667284337
```

DECISION FOREST REGRESSION r2 score: 0.2967831866338452 RMSE : 7.579711887664333 MSE: 57.4520323 MAE: 5.838099999999999

MAPE: 0.7793818583999947

r2 score: 0.9296137881016225 RMSE : 2.470282067199093 MSE: 6.102293491525424 MAE: 2.0082711864406777 MAPE: 0.37432196450623323

In [198]:
```
data = {'y_test': y_test, 'y_pred': y_pred}
```

In [199]:
```
new_df = pd.DataFrame(data)
```

In [200]:
```
diff = (new_df['y_test'] - new_df['y_pred']).abs()
```

In [201]:
```
new_df['diff'] = diff
```

In [202]:
```
new_df = new_df.sort_values(by='diff', ascending=True)
```

In [203]:
```
new_df['y_pred'] = round(new_df['y_pred'], 1)
```

In [204]:
```
new_df
```

Out[204]:

	y_test	y_pred	diff
58	10	10.2	0.171
159	21	21.6	0.643
95	23	22.3	0.703
27	5	3.8	1.184
110	11	12.4	1.436
177	22	20.4	1.563
38	15	13.1	1.882
69	23	21.1	1.889
172	8	10.0	2.018
118	12	14.4	2.394
165	18	20.5	2.508
28	6	8.6	2.552
44	6	8.6	2.558
174	22	19.4	2.603
16	18	15.3	2.664
94	27	24.2	2.768
35	3	5.9	2.932
11	3	6.5	3.511
176	14	10.3	3.659
162	13	16.8	3.841
4	5	8.8	3.848
18	10	5.6	4.359
171	24	19.4	4.569
136	12	16.7	4.660
168	17	11.6	5.350
31	2	7.5	5.541
51	5	10.8	5.841
198	4	9.9	5.926
184	15	8.8	6.157
193	14	20.7	6.741
40	7	13.9	6.881
59	28	21.1	6.907
29	4	11.6	7.630
73	32	24.2	7.773
47	16	7.8	8.178
89	29	20.6	8.436
34	6	15.8	9.761
102	17	7.0	10.017
97	32	21.8	10.182
194	8	20.9	12.918

In [205]:
```
len(new_df)/2
```

Out[205]:
```
20.0
```

In [212]:
```
new_df.iloc[:20]
```

Out[212]:

	y_test	y_pred	diff
58	10	10.2	0.171
159	21	21.6	0.643
95	23	22.3	0.703
27	5	3.8	1.184
110	11	12.4	1.436
177	22	20.4	1.563
38	15	13.1	1.882
69	23	21.1	1.889
172	8	10.0	2.018
118	12	14.4	2.394
165	18	20.5	2.508
28	6	8.6	2.552
44	6	8.6	2.558
174	22	19.4	2.603
16	18	15.3	2.664
94	27	24.2	2.768
35	3	5.9	2.932
11	3	6.5	3.511
176	14	10.3	3.659
162	13	16.8	3.841

In [207]:
```
new_df['diff'].sum()
```

Out[207]:
```
185.154
```

In [208]:
```
new_df['diff'].mean()
```

Out[208]:
```
4.628849999999999
```

In [209]:
```
new_df['diff'].median()
```

Out[209]:
```
3.844500000000001
```

In [210]:
```
# mean = 5.8381
# median = 4.56799
```

In [ ]:
```

```

In [ ]:
```

```