In [13]:
```
import pandas as pd

df = pd.read_csv("robot_inverse_kinematics_dataset.csv")
print("The Dataset is: ")
df
```

```
The Dataset is:
```

Out[13]:

	q1	q2	q3	q4	q5	q6	x	y	z
0	-1.510	-0.763	1.85	-0.817	0.9120	2.320	-0.0947	0.15000	0.301
1	-2.840	0.520	1.58	-1.270	-1.3900	0.617	0.1420	-0.10000	0.225
2	-1.230	0.695	1.22	-1.130	0.0343	6.270	-0.0833	0.22300	0.206
3	-1.990	1.060	1.74	-1.760	-1.2400	4.760	0.1350	-0.03140	0.370
4	1.050	0.836	1.34	-1.890	0.4840	4.380	-0.0560	-0.22900	0.260
...	...	...	...	...	...	...	...	...	...
14995	0.314	-0.534	1.76	1.970	-0.6990	3.870	-0.1130	-0.12800	0.257
14996	2.450	1.360	1.55	2.780	-0.3210	5.310	0.0633	-0.03160	0.450
14997	2.620	1.410	1.56	2.540	1.0600	5.870	0.1310	-0.16000	0.362
14998	-1.890	1.850	1.51	1.090	0.6970	4.070	0.0829	-0.01600	0.441
14999	2.680	-1.790	1.79	2.620	1.5900	2.640	-0.1570	-0.00369	0.254

15000 rows × 9 columns

In [14]:

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.metrics import r2_score
from sklearn.inspection import permutation_importance
import pandas as pd

df = pd.read_csv("robot_inverse_kinematics_dataset.csv")
features = ["x" , "y" , "z"]
X = df[features]
y = [df['q1'] , df['q2'] , df['q3'] , df['q4'] , df['q5'] , df['q6']]

def DT(X , y , testData):
    output = []
    reg = DecisionTreeRegressor(random_state=0)
    reg.fit(X.values , y.values)

    predictedValue = reg.predict(testData)
    output.append(predictedValue[0])

    r2 = r2_score(y , reg.predict(X.values))
    output.append(r2)


    importance = reg.feature_importances_
    for i in range(3):
        output.append(importance[i])

    return output

def RF(X , y , testData):
    output = []
    reg = RandomForestRegressor(random_state=0 , n_estimators=100)
    reg.fit(X.values , y.values)
    predictedValue = reg.predict(testData)
    output.append(predictedValue[0])

    r2 = r2_score(y , reg.predict(X.values))
    output.append(r2)


    importance = reg.feature_importances_
    for i in range(3):
        output.append(importance[i])

    return output
```

In [24]:
```
from IPython.display import display
test = [[-0.0947 , 0.15000 , 0.301]]
result = pd.DataFrame({"Predicted θ1 (rad)" : [DT(X , y[0] , test)[0] , RF(
                       "Predicted θ2 (rad)" : [DT(X , y[1] , test)[0] , RF(
                       "Predicted θ3 (rad)" : [DT(X , y[2] , test)[0] , RF(
                       "Predicted θ4 (rad)" : [DT(X , y[3] , test)[0] , RF(
                       "Predicted θ5 (rad)" : [DT(X , y[4] , test)[0] , RF(
                       "Predicted θ6 (rad)" : [DT(X , y[5] , test)[0] , RF(

df1 = pd.DataFrame({"Real θ1 (rad)" : [y[0][0]] , "Real θ2 (rad)" : [y[1][0
                "Real θ4 (rad)" : [y[3][0]] , "Real θ5 (rad)" : [y[4][0]]

df2 = pd.DataFrame({"Decision tree coefficient of determination (R squared)

                    "Random forest coefficient of determination (R squared)"

result.index = ["Decision Tree" , "Random Forest"]
df1.index = ["Value"]
df2.index = ["For θ1 prediction" , "For θ2 prediction" , "For θ3 prediction
             "For θ5 prediction" , "For θ6 prediction"]
display(df1)
print("-" * 100)
display(result)
print("-" * 100)
display(df2)
```

	Real θ1 (rad)	Real θ2 (rad)	Real θ3 (rad)	Real θ4 (rad)	Real θ5 (rad)	Real θ6 (rad)
Value	-1.51	-0.763	1.85	-0.817	0.912	2.32

	Predicted θ1 (rad)	Predicted θ2 (rad)	Predicted θ3 (rad)	Predicted θ4 (rad)	Predicted θ5 (rad)	Predicted θ6 (rad)
Decision Tree	-1.51000	-0.763000	1.8500	-0.81700	0.912000	2.320000
Random Forest	-1.38802	-0.505554	1.8213	-0.71344	0.660144	2.833229

	Decision tree coefficient of determination (R squared)	Random forest coefficient of determination (R squared)
For θ1 prediction	1.0	0.865935
For θ2 prediction	1.0	0.907527
For θ3 prediction	1.0	0.920667
For θ4 prediction	1.0	0.845902
For θ5 prediction	1.0	0.848114
For θ6 prediction	1.0	0.845358

In [ ]: