Check for MAE, MSE, R2-score, RMSE based on different n_estimators

In [13]:

```
#Check for MAE, MSE, R2-score, RMSE based on different n_estimators
import random as rd
from sklearn.metrics import r2_score, mean_squared_error
i = rd.sample(range(100),10)
x,a,b,c,d = [],[],[],[],[]
for e in i:
    from sklearn.ensemble import RandomForestRegressor
    reg = RandomForestRegressor(n_estimators=e, random_state = 0)
    reg.fit(X,y)
    pred = reg.predict(X)
    x.append(e)
    a.append(np.mean(np.absolute(pred - y)).round(decimals = 2))
    b.append(np.sqrt(mean_squared_error(y, pred)).round(decimals = 2))
    c.append(r2_score(pred , y).round(decimals = 2))
    d.append(np.sqrt(np.mean((pred - y) ** 2)).round(decimals = 2))

    rand_check = pd.DataFrame({
        'n_estimators': np.array(x).flatten(),
        'MAE': np.array(a).flatten(),
        'MSE': np.array(b).flatten(),
        'R2-Score': np.array(c).flatten(),
        'RMSE': np.array(d).flatten(),
    })
rand_check
```

Out[13]:

	n_estimators	MAE	MSE	R2-Score	RMSE
0	35	26814.29	73346.44	0.89	73346.44
1	9	33000.00	60696.57	0.93	60696.57
2	80	26437.50	65478.66	0.92	65478.66
3	81	27283.95	67474.30	0.91	67474.30
4	36	26430.56	71351.60	0.90	71351.60
5	60	25658.33	66271.58	0.92	66271.58
6	63	25682.54	65660.94	0.92	65660.94
7	75	26073.33	66626.38	0.92	66626.38
8	91	26159.34	69082.49	0.91	69082.49
9	95	26915.79	71190.65	0.90	71190.65

Check for minimum RMSE and returning best best n_estimator

In [14]:

```
#Check for minimum RMSE and returning best best n_estimator
#p = np.where(d == np.amin(d))
p= d.index(min(d))
print("Best n_estimator value: ", x[p])
evaluation = pd.DataFrame({
    'n_estimators': np.array(x[p]).flatten(),
    'MAE': np.array(a[p]).flatten(),
    'MSE': np.array(b[p]).flatten(),
    'R2-Score': np.array(c[p]).flatten(),
    'RMSE': np.array(d[p]).flatten(),
})
evaluation
```

```
Best n_estimator value:  9
```

Out[14]:

	n_estimators	MAE	MSE	R2-Score	RMSE
0	9	33000.0	60696.57	0.93	60696.57