n_estimators

- n_estimetore are te numbers of trees in RandomForest, how many subsets of data you want to make for your model?
- Its better to take more no of trees it's better for good prediction, but it also depends on our running laptop,cpu how many we should take.
- Its default value is hundred
- In version 0.22 its defualt value is 10 - 100

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('Salary_data.csv')
df.head()
```

Out[ ]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [ ]:
```
X = df.iloc[: , :-1]
y = df.iloc[: ,-1:]
```

In [ ]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test, = train_test_split(X, y, test_size= 0.2)
regressor= RandomForestRegressor(n_estimators=90, random_state=42,max_depth = 3, )
regressor.fit(X_train,y_train)
regressor.predict([[65]])
```

```
C:\Users\Faiza\AppData\Local\Temp/ipykernel_10344/979841258.py:5: DataConversionWarni
ng: A column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples,), for example using ravel().
  regressor.fit(X_train,y_train)
C:\Users\Faiza\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\bas
e.py:450: UserWarning: X does not have valid feature names, but RandomForestRegressor
was fitted with feature names
  warnings.warn(
```

Out[ ]:
```
array([120512.50518519])
```

In [ ]:
```
regressor.predict([[56],[56],[56]])
```

```
C:\Users\Faiza\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\bas
e.py:450: UserWarning: X does not have valid feature names, but RandomForestRegressor
was fitted with feature names
  warnings.warn(
```

Out[ ]:
```
array([120512.50518519, 120512.50518519, 120512.50518519])
```

In [ ]:
```
score=regressor.score(X_test, y_test)
print("The Accuracy of model is =", score)
```

```
The Accuracy of model is = 0.9127421628483616
```

In [ ]:
```
# Scatter
plt.scatter(X, y, color = 'red')
plt.plot(X, regressor.predict(X),
         color = 'green')
plt.title('Random Forest Regression')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.show()
```