In [31]:

```
import pandas as pd# we import pandas to handle the file and its save ou
import numpy as np #for creating arrays
import seaborn as sns #for data visualzation
import matplotlib.pyplot as plt # for data visualization

df1 = pd.read_csv(r'C:\Users\PC-chetan\Desktop\train.csv') # trian data

df2 = pd.read_csv(r'C:\Users\PC-chetan\Desktop\test.csv') # test data

df1.education.fillna("Bachelor's", inplace=True)

df2.education.fillna("Bachelor's", inplace=True)

df1.previous_year_rating.fillna('3.0',inplace=True)

df2.previous_year_rating.fillna('3.0',inplace=True)

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

df1.drop(columns=['employee_id','region','recruitment_channel'], inplace

df2.drop(columns=['employee_id','region','recruitment_channel'], inplace

#lets encode the education in their degree of importance
df1['education'] = df1['education'].replace(("Master's & above", "Bachel
                                                (3, 2, 1))
df2['education'] = df2['education'].replace(("Master's & above", "Bachel

df1.gender = le.fit_transform(df1.gender)

df1.department = le.fit_transform(df1.department)

df2.department = le.transform(df2.department)


df2['gender'] = df2['gender'].replace(("m", "f"),(1,0))
```

In [32]:

```
df1.shape
```

Out[32]:

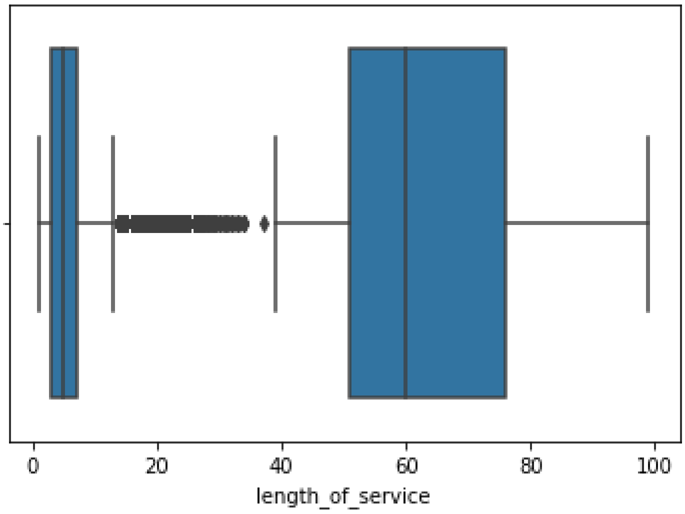
```
(54808, 10)
```

In [33]:

```
df1.select_dtypes('number').head()

df2.select_dtypes('number').head()

sns.boxplot(data=df1,x=df1['avg_training_score'])

df1.shape

Q1=df1['avg_training_score'].quantile(0.25)
Q3=df1['avg_training_score'].quantile(0.75)
IQR=Q3-Q1
print(Q1)
print(Q3)
print(IQR)
min_1 = Q1-(1.5)*IQR
max_1 = Q3+(1.5)*IQR
```

```
print(min_1)
print( max_1)


df1['avg_training_score'].unique()

df1 = df1[df1['avg_training_score']< max_1]

df1.shape


sns.boxplot(data=df1,x=df1['length_of_service'])

Q2=df1['length_of_service'].quantile(0.25)
Q4=df1['length_of_service'].quantile(0.75)
IQRt=Q4-Q2
print(Q2)
print(Q4)
print(IQRt)
min_2 = Q2-1.5*IQRt
max_2 = Q4+1.5*IQRt
print(max_2,min_2)

df1['length_of_service'].unique()
df1 = df1[df1['length_of_service'] > 13]
```

```
51.0
76.0
25.0
13.5
113.5
3.0
7.0
4.0
13.0 -3.0
```



In [34]:

```
# feature engineering
#it is the most important part of the data preprocessing
```

In [35]:

```
df1.shape
```

Out[35]:

```
(3489, 10)
```

In [36]:
```
df1['sum_metric'] = df1['awards_won?']+ df1['previous_year_rating']


# creating a total score column
df1['total_score'] = df1['avg_training_score'] * df1['no_of_trainings']
```

In [37]:
```
pd.set_option('display.max_rows', 5000) # for getting the max veiw of ra
pd.set_option('display.max_column', 5000) # for getting the max veiw of
```

In [38]:
```
df1[(df1['previous_year_rating'] == 1.0) &
      (df1['awards_won?'] == 0) & (df1['avg_training_score'] < 60) & (df
```

Out[38]:

	department	education	gender	no_of_trainings	age	previous_year_rating	length_of
11803	7	2	1	1	42	1.0	
40379	4	3	1	1	46	1.0	

In [39]:
```
df1 = df1.drop(df1[(df1['previous_year_rating'] == 1.0) &
      (df1['awards_won?'] == 0) & (df1['avg_training_score'] < 60) & (df
```

In [40]:
```
df1.shape
```

Out[40]:
```
(3487, 12)
```

In [41]:
```
y = df1['is_promoted']
x = df1.drop(columns=['is_promoted'])
```

In [74]:
```
#X_train, X_test, y_train, y_test =train_test_split(x,test_size=.3)

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size= 0.3,
```

In [75]:
```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
```

In [76]:
```
dtree.fit(x_train,y_train)
```

Out[76]:
```
DecisionTreeClassifier()
```

In [96]:
```
dtree.score(x_test,y_test)
```

Out[96]:
```
0.9130850047755492
```

In [90]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 40)
```

In [91]:
```
rf.fit(x_train,y_train)
```

Out[91]:
```
RandomForestClassifier(n_estimators=40)
```

In [92]:
```
rf.predict(x_test)
```

Out[92]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [95]:
```
rf.score(x_train,y_train)
```

Out[95]:
```
0.9967213114754099
```