DVA_lab-7_205229118_Mahalakshmi.S

April 28, 2021

0.0.1 Lab7. Data Visualization in Seaborn

[1]:
```
# Import necessary packages
import pandas as pd
import csv
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

0.0.2 Import train_upvote_mini.csv file

[2]:
```
df = pd.read_csv("train_upvote_mini.csv")
df.head()
```

[2]:

	ID	Tag	Reputation	Answers	Username	Views	Upvotes
0	52664	a	3942.0	2.0	155623	7855.0	42.0
1	327662	a	26046.0	12.0	21781	55801.0	1175.0
2	468453	c	1358.0	4.0	56177	8067.0	60.0
3	96996	a	264.0	3.0	168793	27064.0	9.0
4	131465	c	4271.0	4.0	112223	13986.0	83.0

0.0.3 What is its size?

[3]:
```
df.shape
```

[3]:
```
(15440, 7)
```

0.0.4 Show the types of each feature

[4]:
```
df.dtypes
```

[4]:
```
ID              int64
Tag            object
Reputation    float64
Answers       float64
Username        int64
Views         float64
```

```
Upvotes       float64
dtype: object
```

0.0.5 How many unique "tag" available?

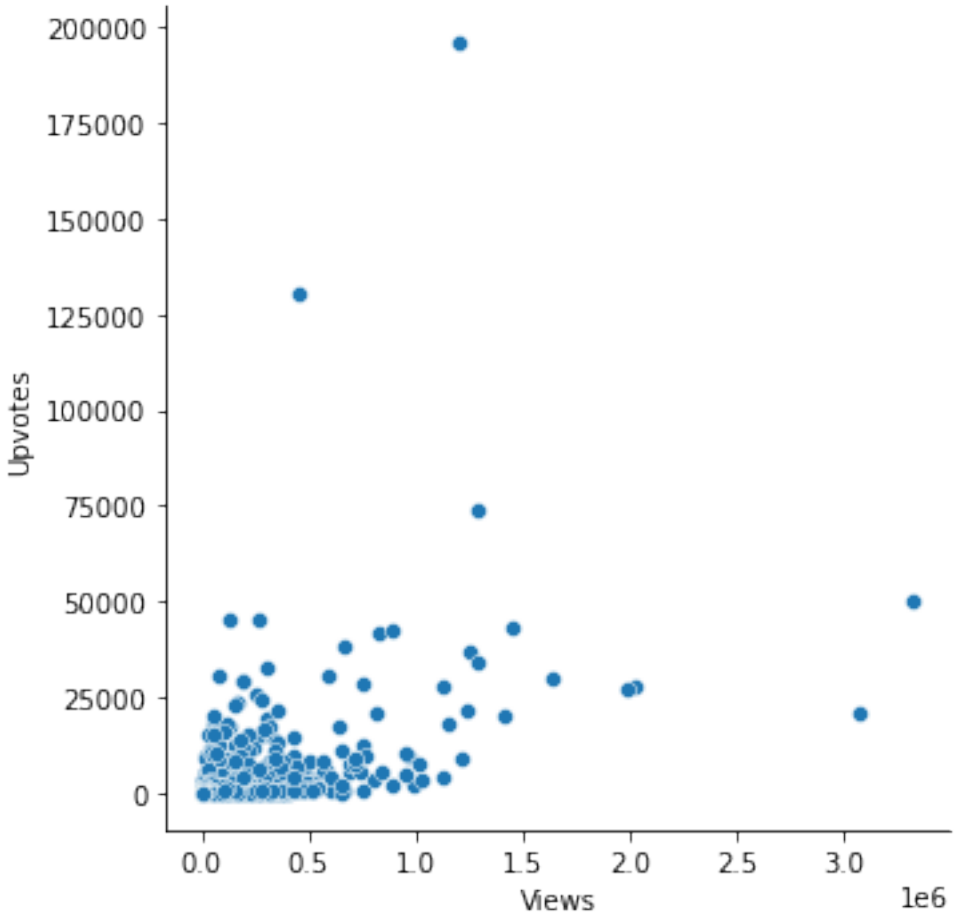
[5]:
```
df.Tag.nunique()
```

[5]:
```
10
```

0.0.6 Show scatterplot (inherited from matplotlib) and relplot between "views" and "upvotes"

[10]:
```
sns.relplot(x="Views", y="Upvotes", data = df)
```

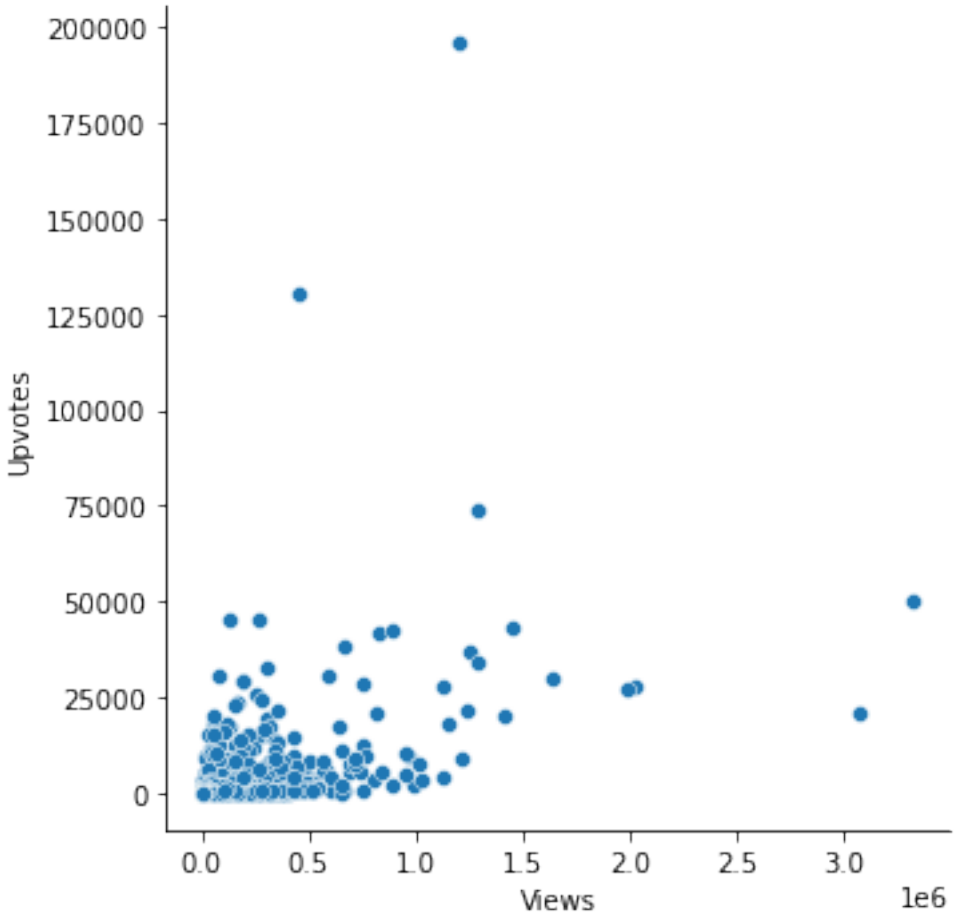
[10]:
```
<seaborn.axisgrid.FacetGrid at 0x1e178852550>
```



0.0.7 Plot replot between "Views" and "Upvotes"

[16]:
```
sns.relplot(x="Views", y="Upvotes", data = df)
```

[16]: <seaborn.axisgrid.FacetGrid at 0x1e1057eb430>

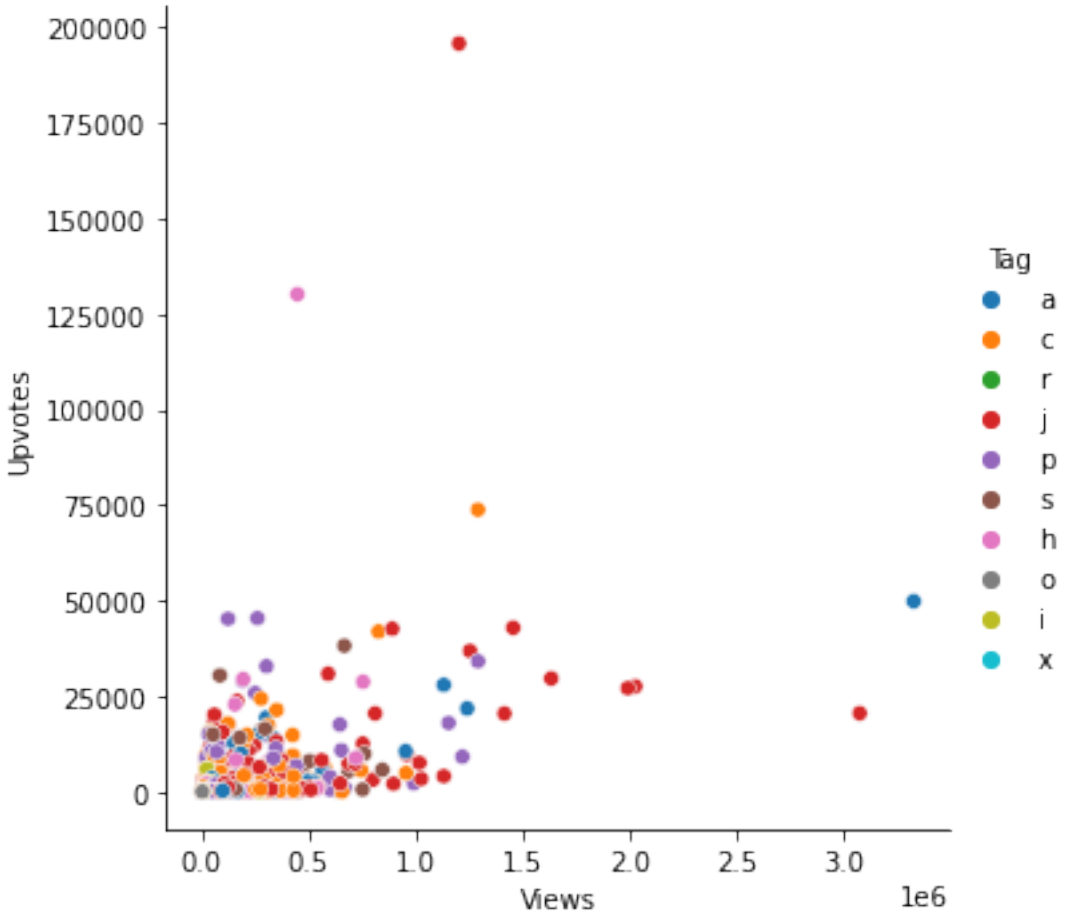


Next, we want to see the tag associated with data.

0.0.8 Plot relplot between "Views" and "Upvotes" with hue as "Tag"

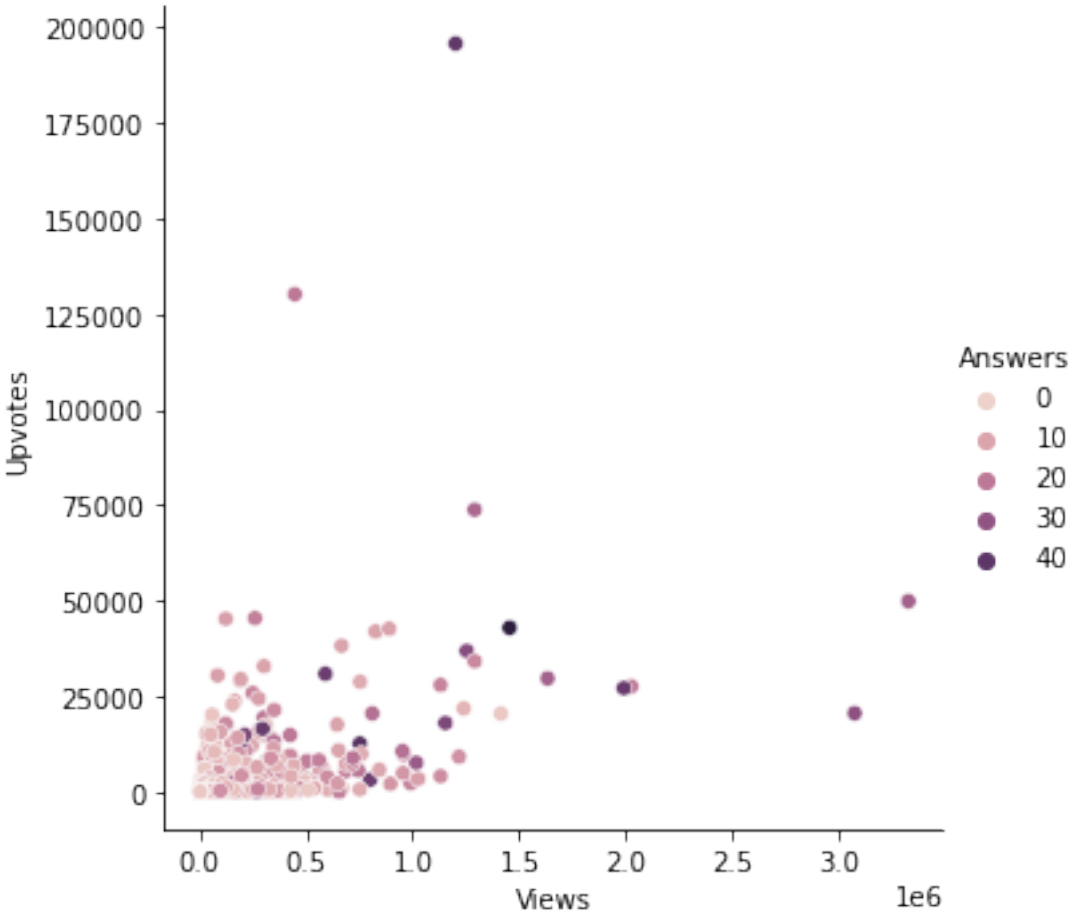
[11]:
```
sns.relplot(x="Views", y="Upvotes", hue = "Tag", data = df)
```

[11]: <seaborn.axisgrid.FacetGrid at 0x1e10238d6a0>



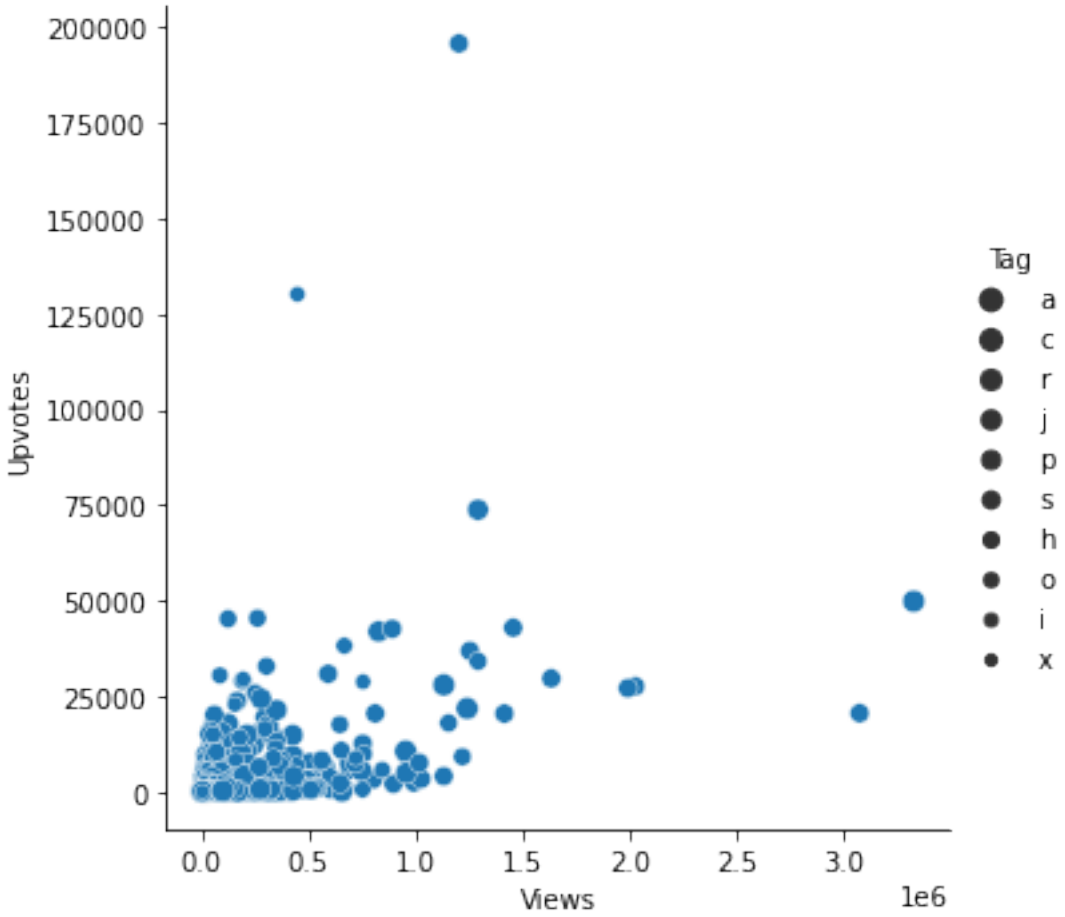
0.0.9 Plot relplot between "Views" and "Upvotes" with hue as "Answers"

[12]:
```
sns.relplot(x="Views", y="Upvotes", hue = "Answers", data = df);
```



0.0.10 Plot relplot between "Views" and "Upvotes" with size as "Tag"

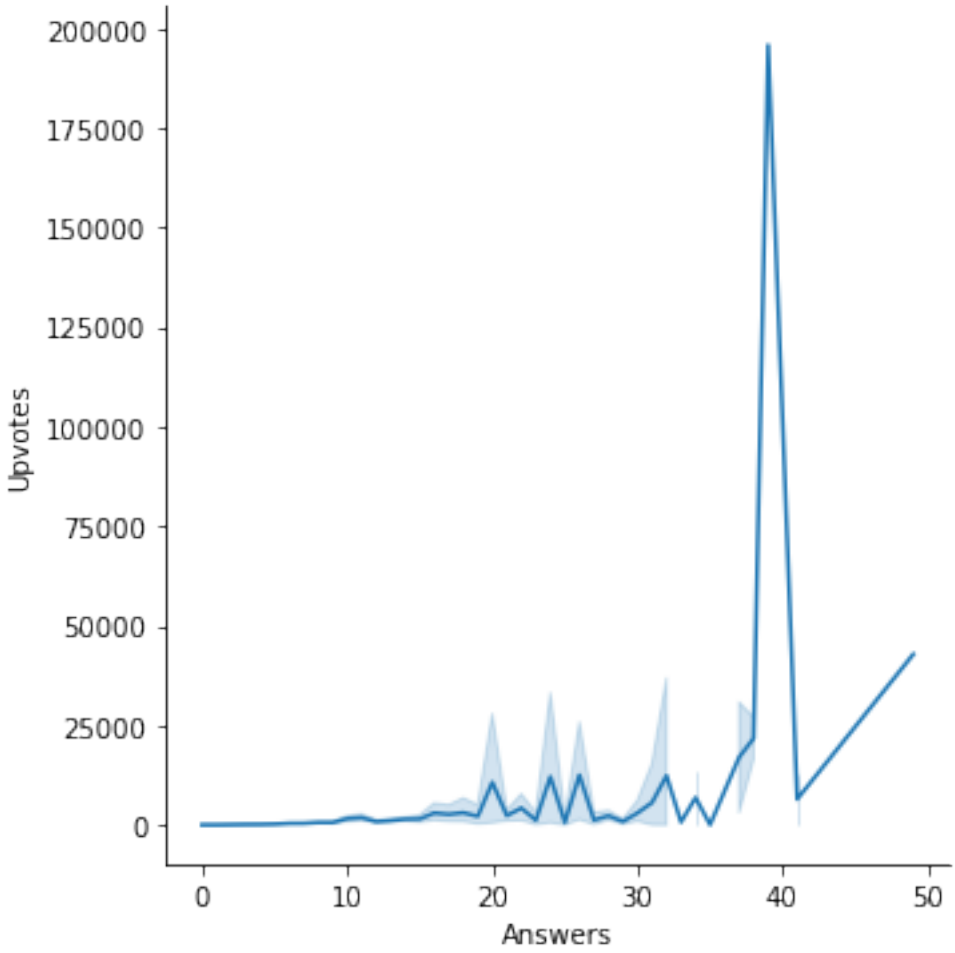
[13]:
```
sns.relplot(x="Views", y="Upvotes", size = "Tag", data = df);
```



Does no of times question answered impact the no. of upvotes?

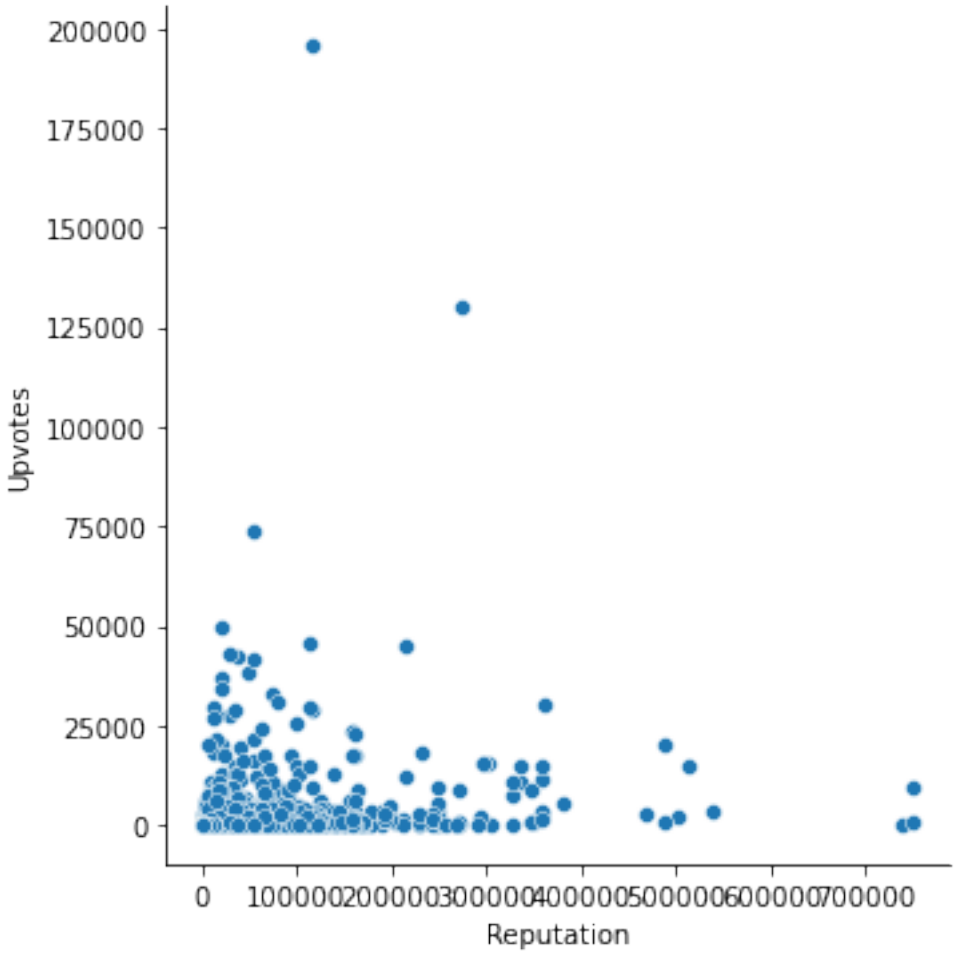
0.0.11 Plot line chart using relplot between "Answers" and "Upvotes"

[6]:
```
sns.relplot(data=df, x='Answers', y='Upvotes', kind='line')
plt.show()
```



0.0.12 Does Reputation score of question author impact no of upvotes?. Draw replot.

[7]:
```
sns.relplot(data=df, x='Reputation', y='Upvotes')
plt.show()
```



0.0.13 Jitter Plot

For jitter plot we'll be using another dataset from the problem HR analysis challenge, let's import the dataset now.

[15]:
```
df2 = pd.read_csv("train_hr_mini.csv")
df2.head()
```

[15]:

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating
0	65438	Sales & Marketing	region_7	Master's & above	f	sourcing	1	35	5.0
1	65141	Operations	region_22	Bachelor's	m				
2	7513	Sales & Marketing	region_19	Bachelor's	m				
3	2542	Sales & Marketing	region_23	Bachelor's	m				
4	48945	Technology	region_26	Bachelor's	m				

```
1               other               1   30                  5.0
2           sourcing                1   34                  3.0
3               other               2   39                  1.0
4               other               1   45                  3.0

   length_of_service  KPIs_met >80%  awards_won?  avg_training_score  \
0                  8              1            0                  49
1                  4              0            0                  60
2                  7              0            0                  50
3                 10              0            0                  50
4                  2              0            0                  73

   is_promoted
0            0
1            0
2            0
3            0
4            0
```

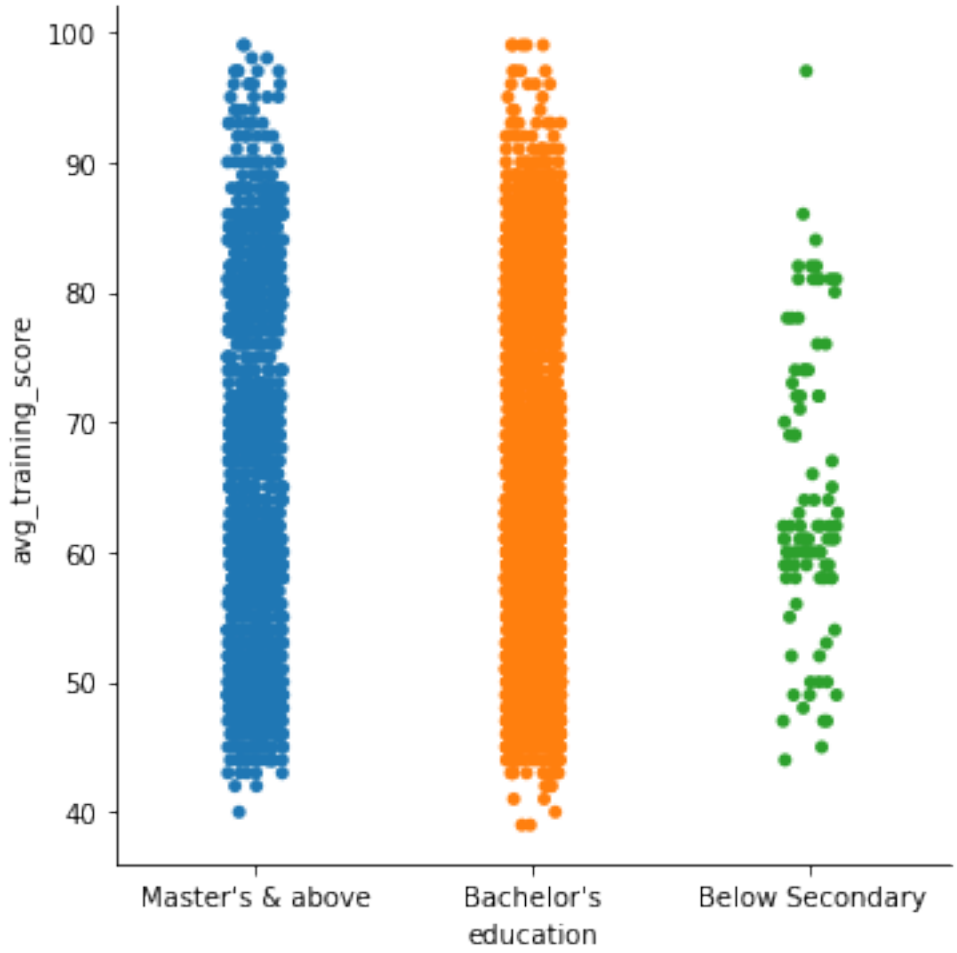
[18]:
```
df2.shape
```

[18]:
```
(6397, 14)
```

0.0.14 Show Jitter plot between education and avg_training_score

[17]:
```
sns.catplot(x="education", y="avg_training_score", data=df2)
```

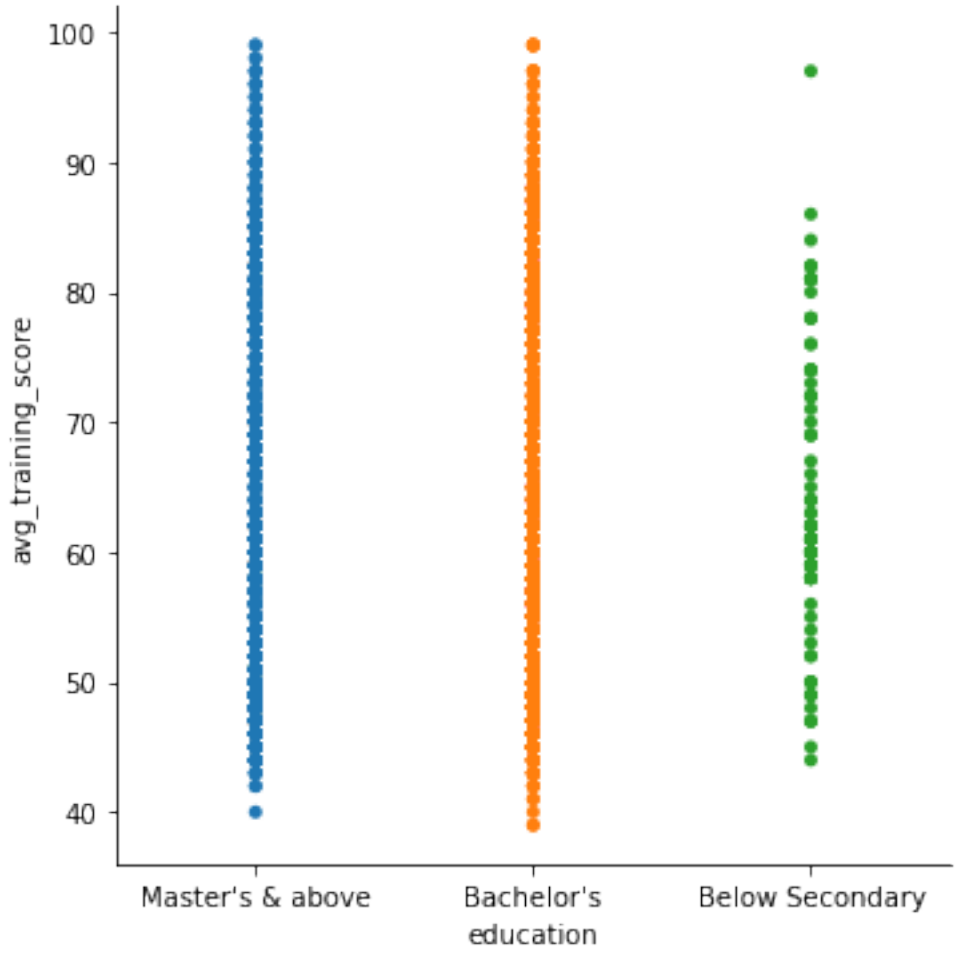
[17]:
```
<seaborn.axisgrid.FacetGrid at 0x1e106cea040>
```



Here, there are a lot of deviation from true values of the points that is called Jitter. So, let us make Jitter to false and visualize data. ### Show the Jitter Plot

[19]:
```
sns.catplot(x="education", y="avg_training_score", jitter = False,  data=df2)
```

[19]:
```
<seaborn.axisgrid.FacetGrid at 0x1e1069a07f0>
```



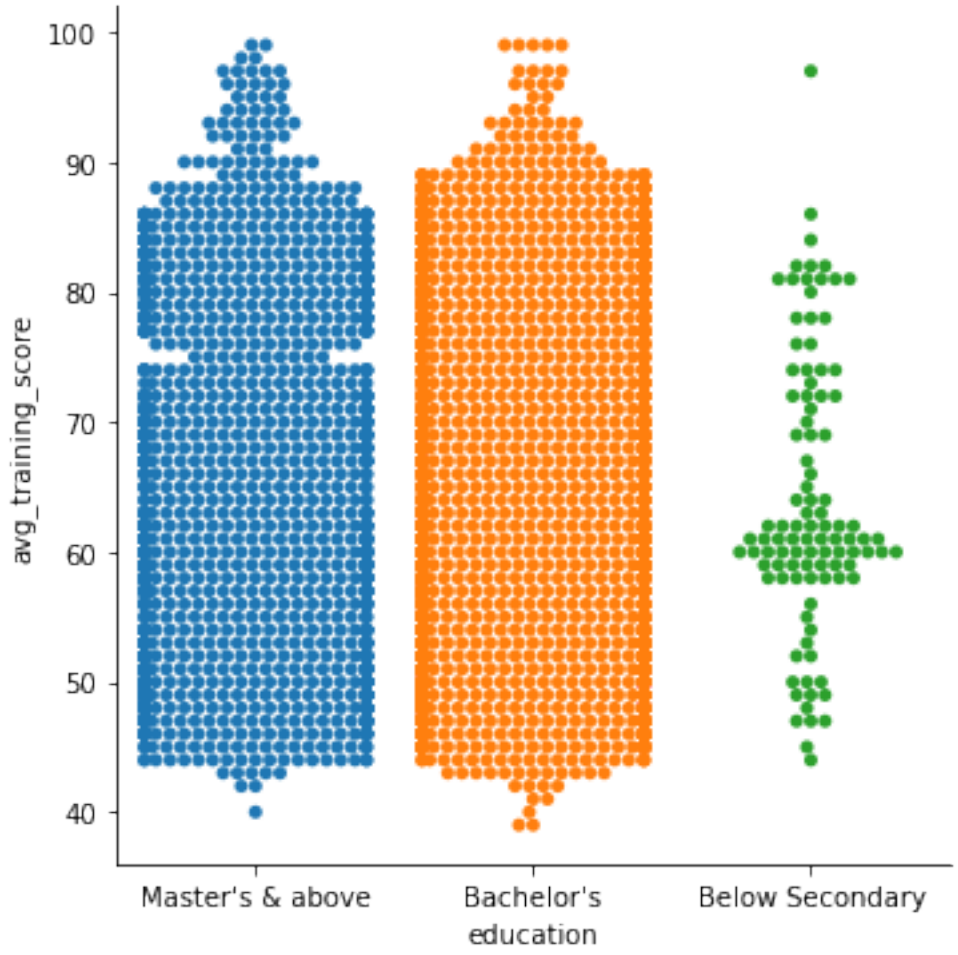
0.0.15 Sworm Plot

Swarm plot adjusts the points along the categorical axis using an algorithm that prevents them from overlapping. It can give a better representation of the distribution of observations. ### Plot Swarm plot between education category and avg_training_score

[20]:
```
sns.catplot(x="education", y="avg_training_score", kind = "swarm", data=df2)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 56.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 81.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

[20]:
```
<seaborn.axisgrid.FacetGrid at 0x1e105860c70>
```

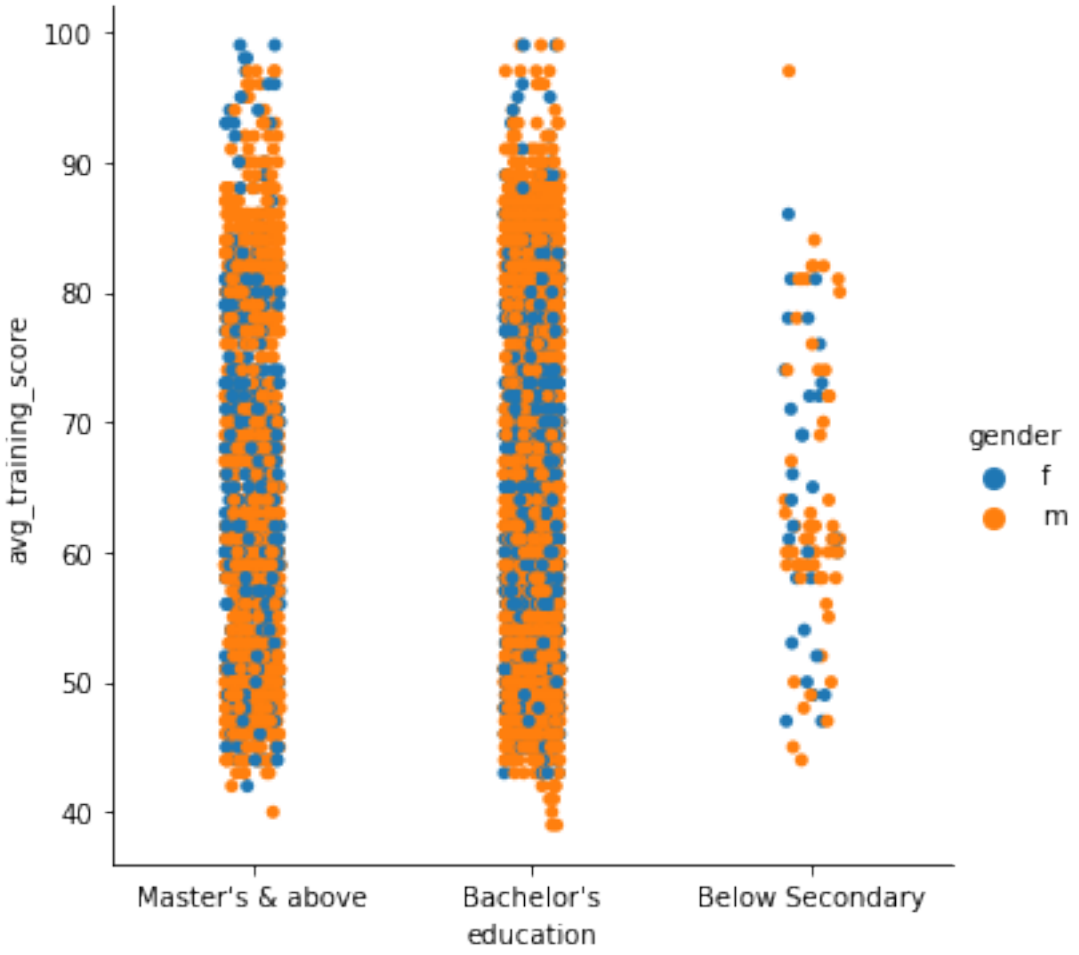


0.0.16 Hue Plot

Now we want to introduce another variable or another dimension in our plot, we can use the hue parameter. We want to see the gender distribution in the plot of education category and avg_training_score ### Show Hue Plot to see the gender distribution in the plot of education category and avg_training_score.Here, hue is "gender".

[21]:
```
sns.catplot(x="education", y="avg_training_score", hue = "gender", data=df2)
```

[21]:
```
<seaborn.axisgrid.FacetGrid at 0x1e106d85eb0>
```

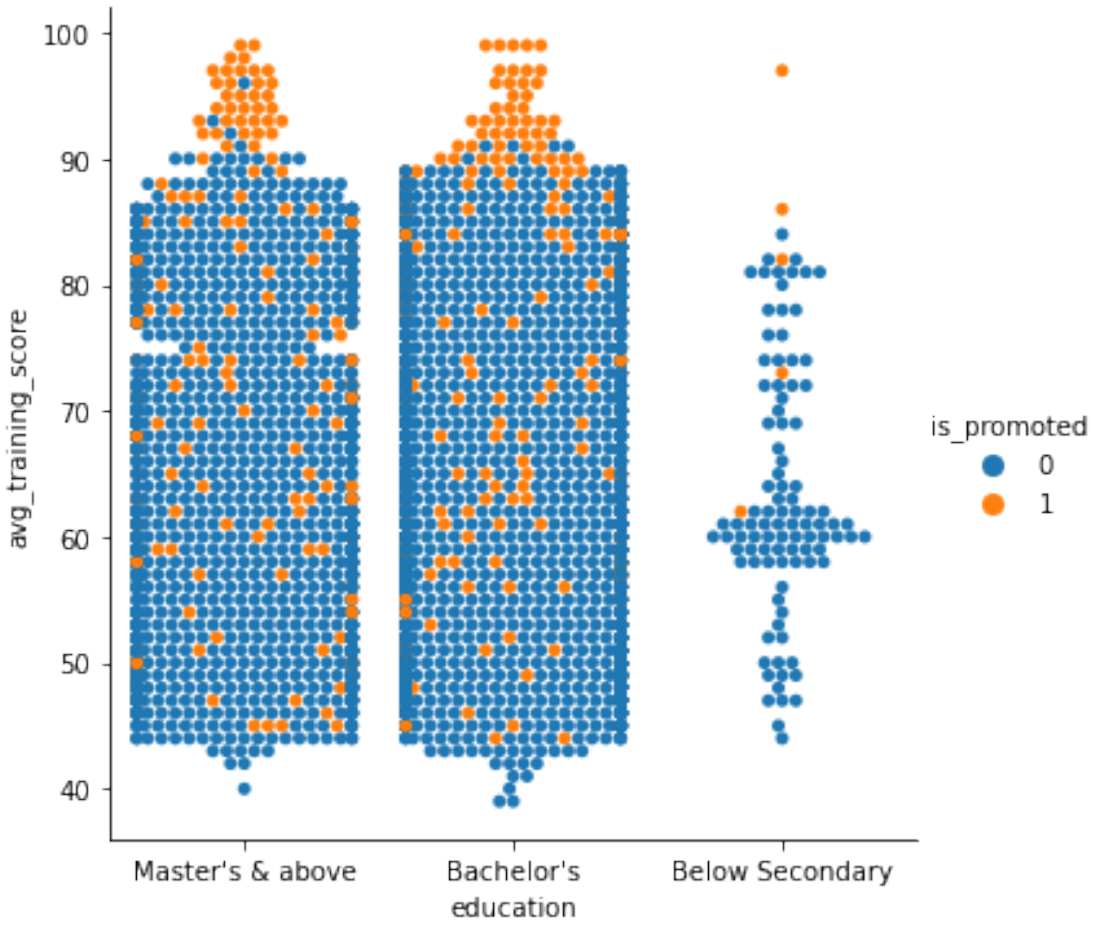


0.0.17 Who are all promoted considering education and avg training score?. Draw swarm plot with hue as "is_promoted"

[22]:
```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind =␣
↪"swarm", data=df2)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 56.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 81.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

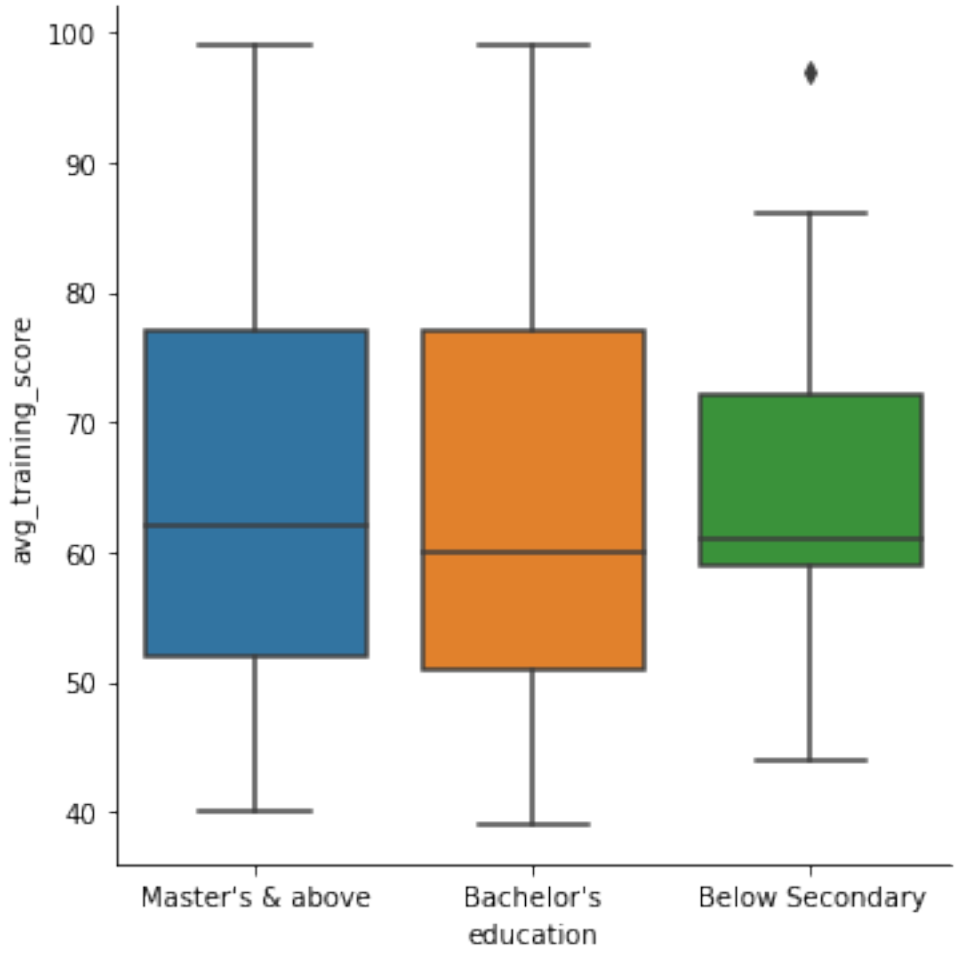
[22]:
```
<seaborn.axisgrid.FacetGrid at 0x1e106cb0fd0>
```



From this plot, we can clearly see people with higher scores got a promotion. ### Box Plot Boxplot shows three quartile values of the distribution along with the end values. Each value in the boxplot corresponds to actual observation in the data. ### Draw box plot between education and avg_training_score

[29]:
```
sns.catplot(x="education", y="avg_training_score", kind = "box", data=df2)
```

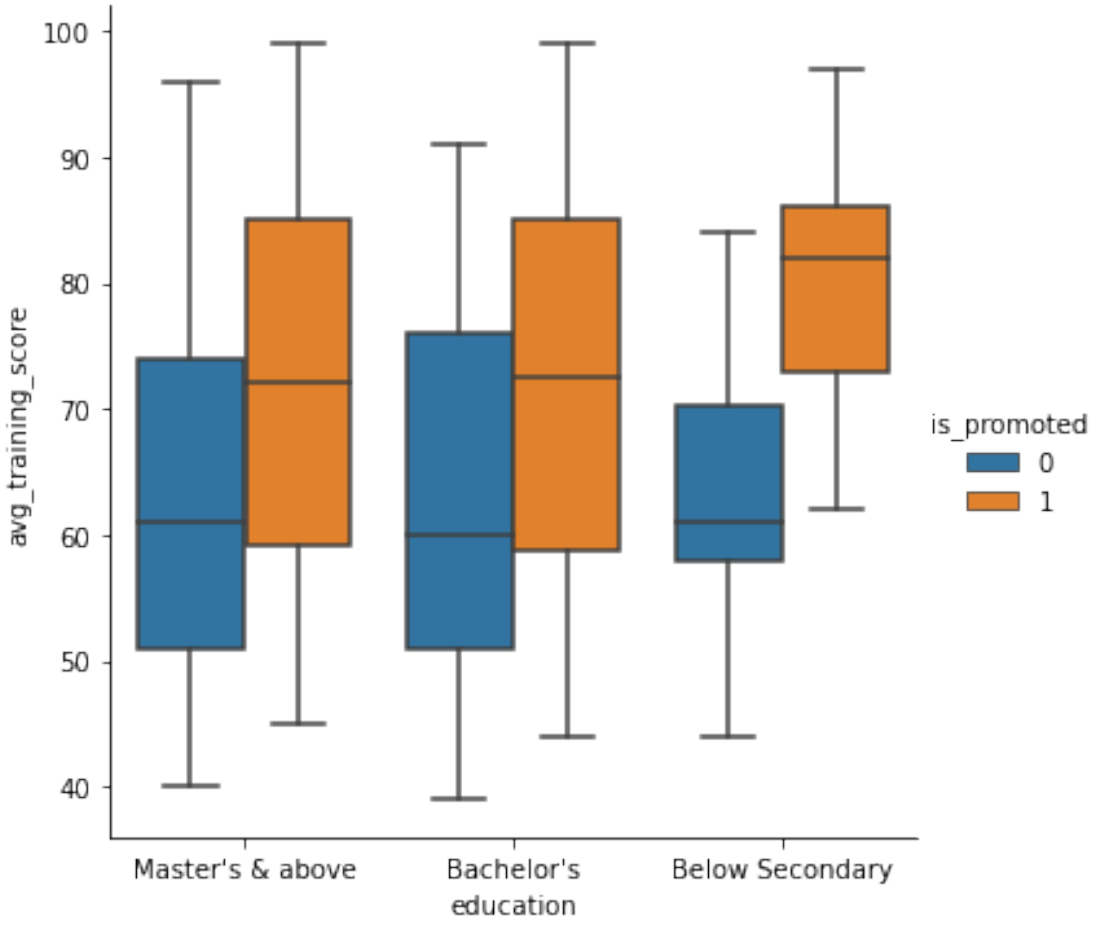
[29]:
```
<seaborn.axisgrid.FacetGrid at 0x1e108e68ac0>
```



From this chart, we can understand that promotees with masters degree have a minimum of 40, maximum of 100 scores and average score of around 62. Similarly, we can see the 25th and 75th percetile scores are around 52 and 78. Similarly, we can interpret for bachelors and below secondary categories as well. ### Box Plot with Hue Dimension ### Who are promoted and not promoted considering education and avg_training_score?. Draw Box Plot.

[30]:
```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind =␣
↪"box", data=df2)
```

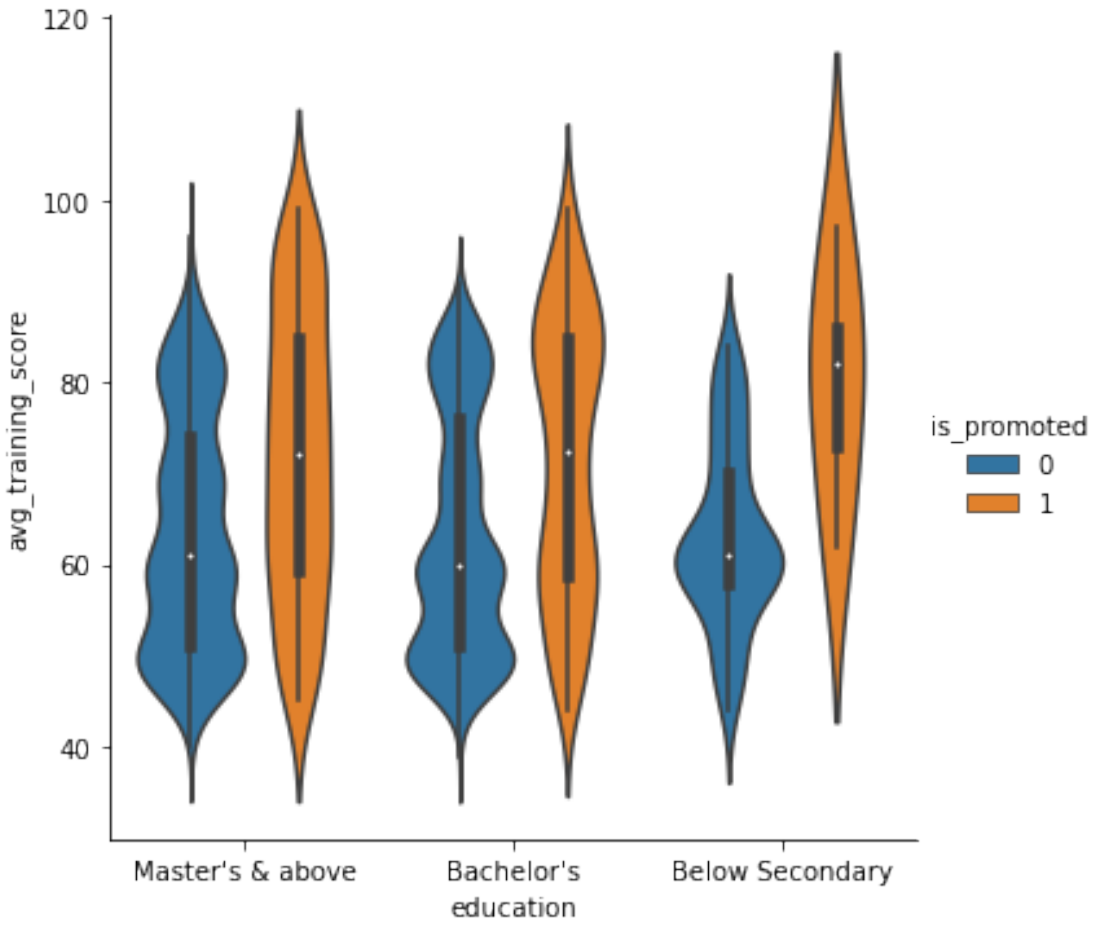
[30]:
```
<seaborn.axisgrid.FacetGrid at 0x1e106b1fa90>
```



From this figure, We can understand 5 types of percentile scores of candidates who are promoted or not with various education levels. Candidates with master degree and avg training score value of 74 approx have been promoted in the past. ### Violin Plot The violin plots combine the boxplot and kernel density estimation procedure to provide richer description of the distribution of values. The quartile values are displayed inside the violin. ### Show violin plot between education categories and avg training score with hue as "is_promoted" target variable

[31]:
```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind =␣
↪"violin", data=df2)
```

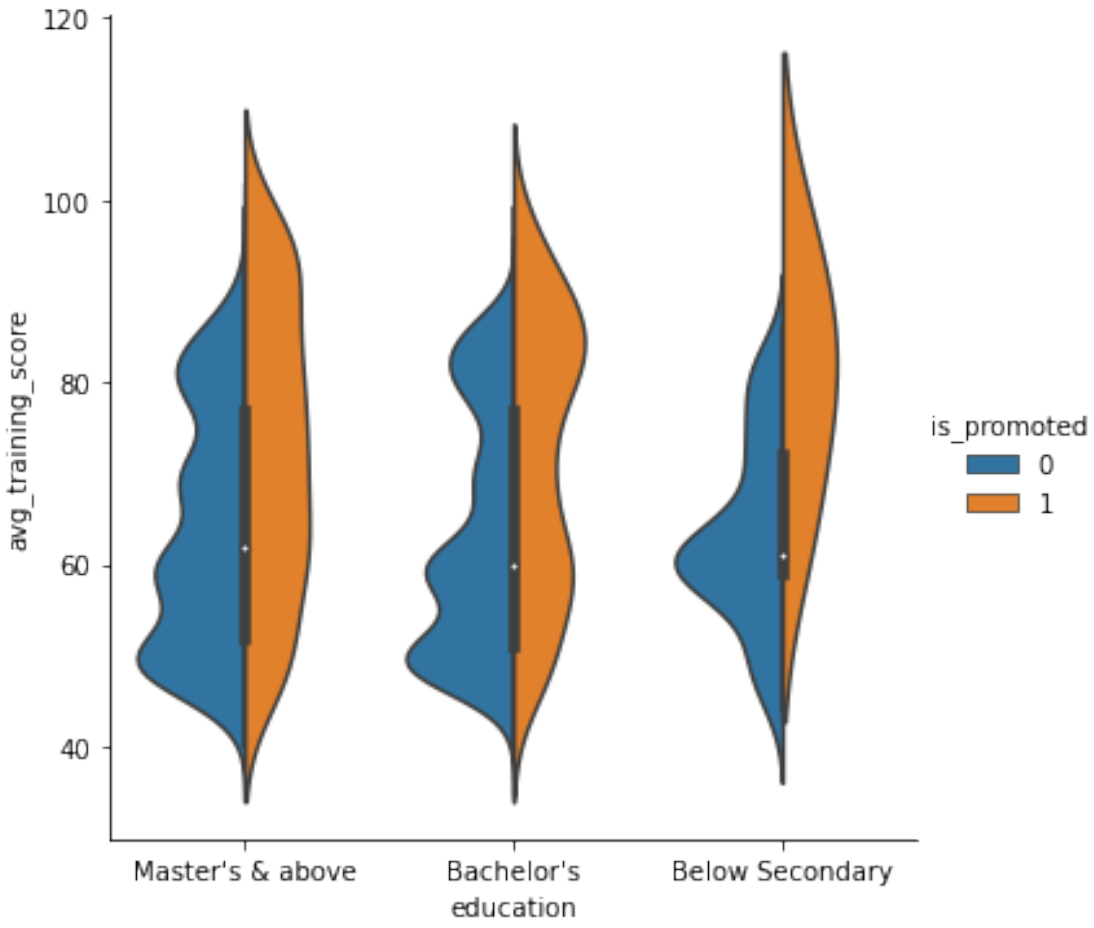
[31]:
```
<seaborn.axisgrid.FacetGrid at 0x1e108e68880>
```



We can see in the above violin plot that each education category is represented with 2 violins one for promoted and the other not promoted target. We can also split the violin when the hue semantic parameter has only two levels, which could also be helpful in saving space on the plot. ### Draw Violin plot with only 2 hue levels, use split attribute

[32]:
```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind =␣
↪"violin", split = True, data=df2)
```

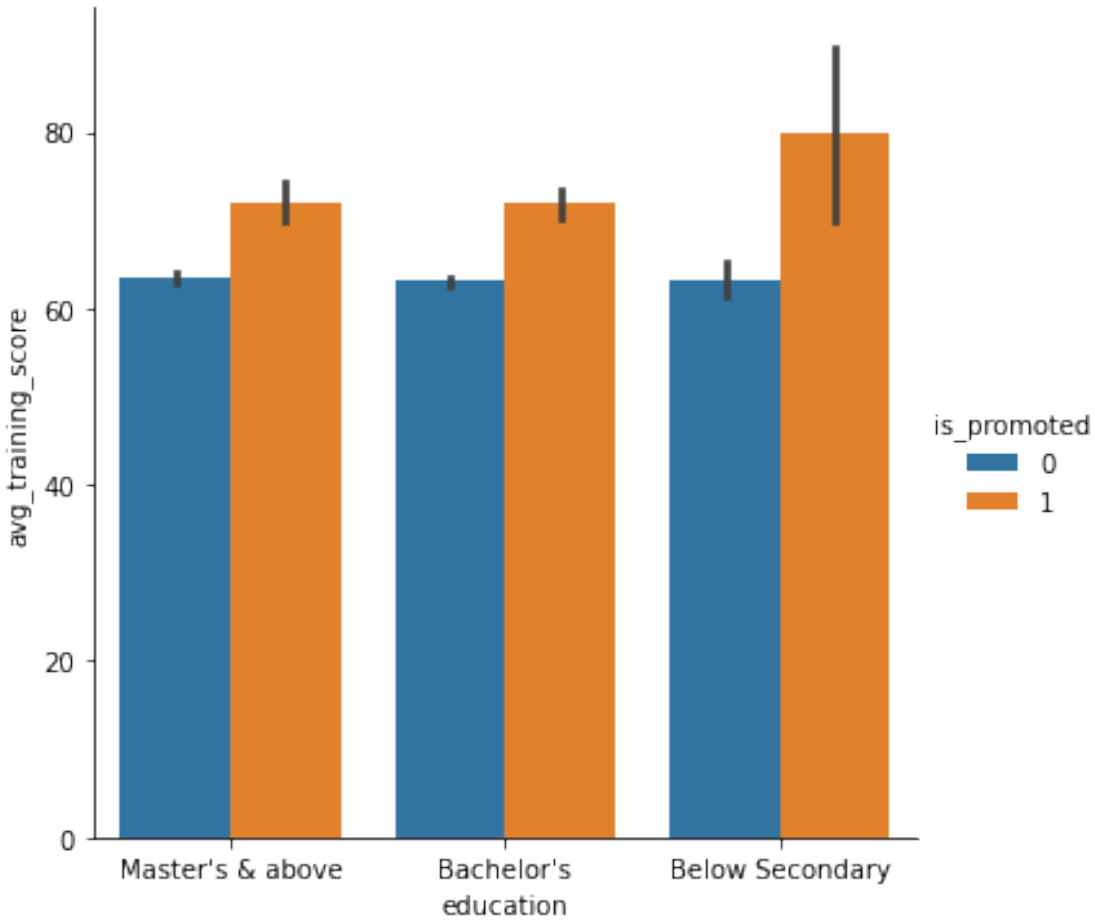
[32]:
```
<seaborn.axisgrid.FacetGrid at 0x1e108fe40a0>
```



0.0.18 Using catplot(), draw a Bar Chart between "education" and "avg_training_score", with hue as "is_promoted"

[33]:
```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind =␣
↪"bar", data=df2)
```

[33]:
```
<seaborn.axisgrid.FacetGrid at 0x1e1058336d0>
```

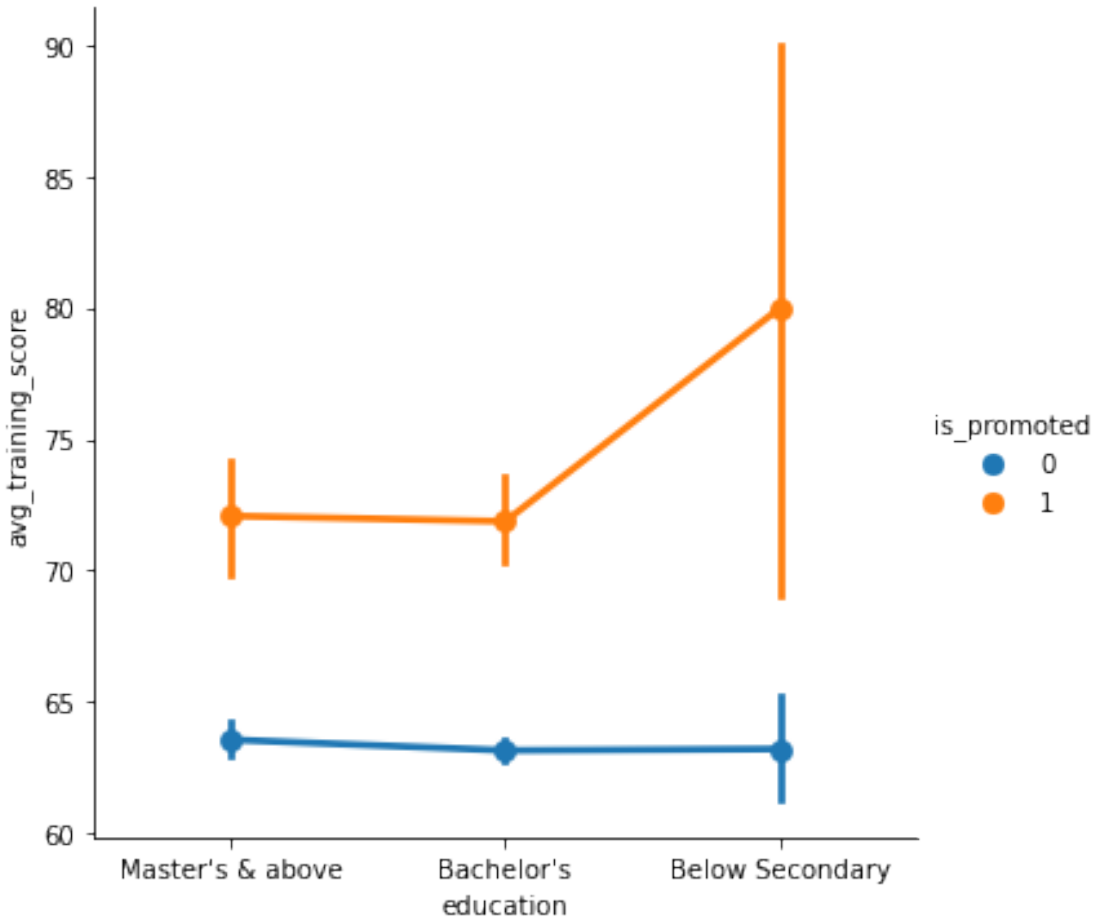


0.0.19 Point Plot

Point plot points out the estimate value and confidence interval. Pointplot connects data from the same hue category. This helps to identify how the relationship is changing in a particular hue category. ### Show point plot between education and avg training score with hue promotion category

[34]:
```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind =␣
↪"point", data=df2)
```

[34]:
```
<seaborn.axisgrid.FacetGrid at 0x1e1058ed730>
```



In the above figure, candidates with higher average training score are promoted. Since, we have taken mini dataset with around 700 samples, confidence interval is high for below secondary education level. Graph will show better plot if we take full dataset. ### Multiple Dimension in Seaborn So far, we have introduced 3 dimensions. Now, let us introduce another dimension, gender, in our plot. We can use Swarm plot to represent is_promoted attribute as hue and gender attribute as a faceting variable. ### Draw swarm plot for education, avg training score, hue as is_promoted for male and female category

[35]:
```
sns.catplot(x="education", y="avg_training_score", hue="is_promoted",
            col="gender", aspect=.9,
            kind="swarm", data=df2);
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 10.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 46.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
```

```
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 44.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 76.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

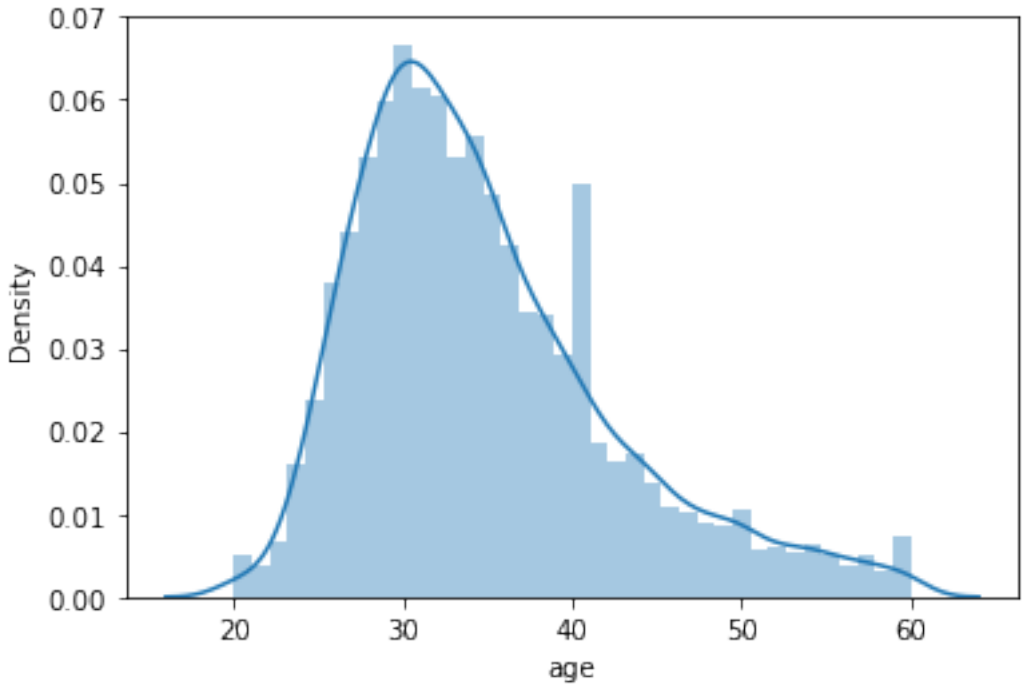
0.0.20 Plot Univariate Distributions

Plot Histogram with kernel density estimate value for age attribute

[36]:
```
sns.distplot(df2.age)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[36]:
```
<AxesSubplot:xlabel='age', ylabel='Density'>
```

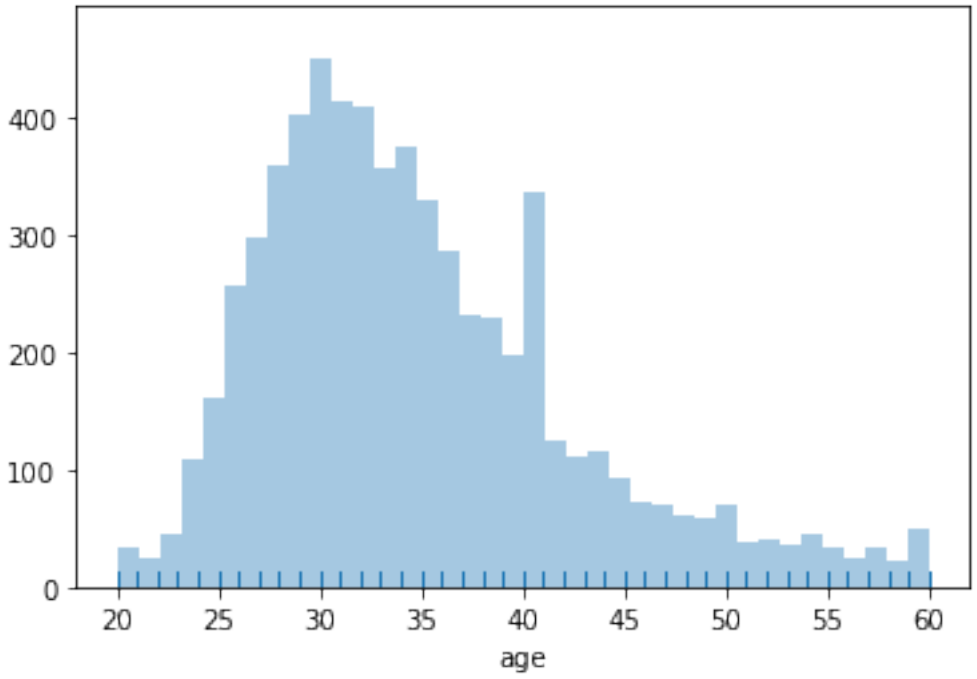


We can understand from .this plot, the average age of candidates. Most of the promotion candidates have age around 25 to 35 years.KDE plot encodes the density of observations (ie., age) on one axis with height along the other axis. ### Show only Histogram for age variable, without KDE

[37]:
```
sns.distplot(df2.age, kde=False, rug = True)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2055:
FutureWarning: The `axis` variable is no longer used and will be removed.
Instead, assign variables directly to `x` or `y`.
  warnings.warn(msg, FutureWarning)
```

[37]:
```
<AxesSubplot:xlabel='age'>
```

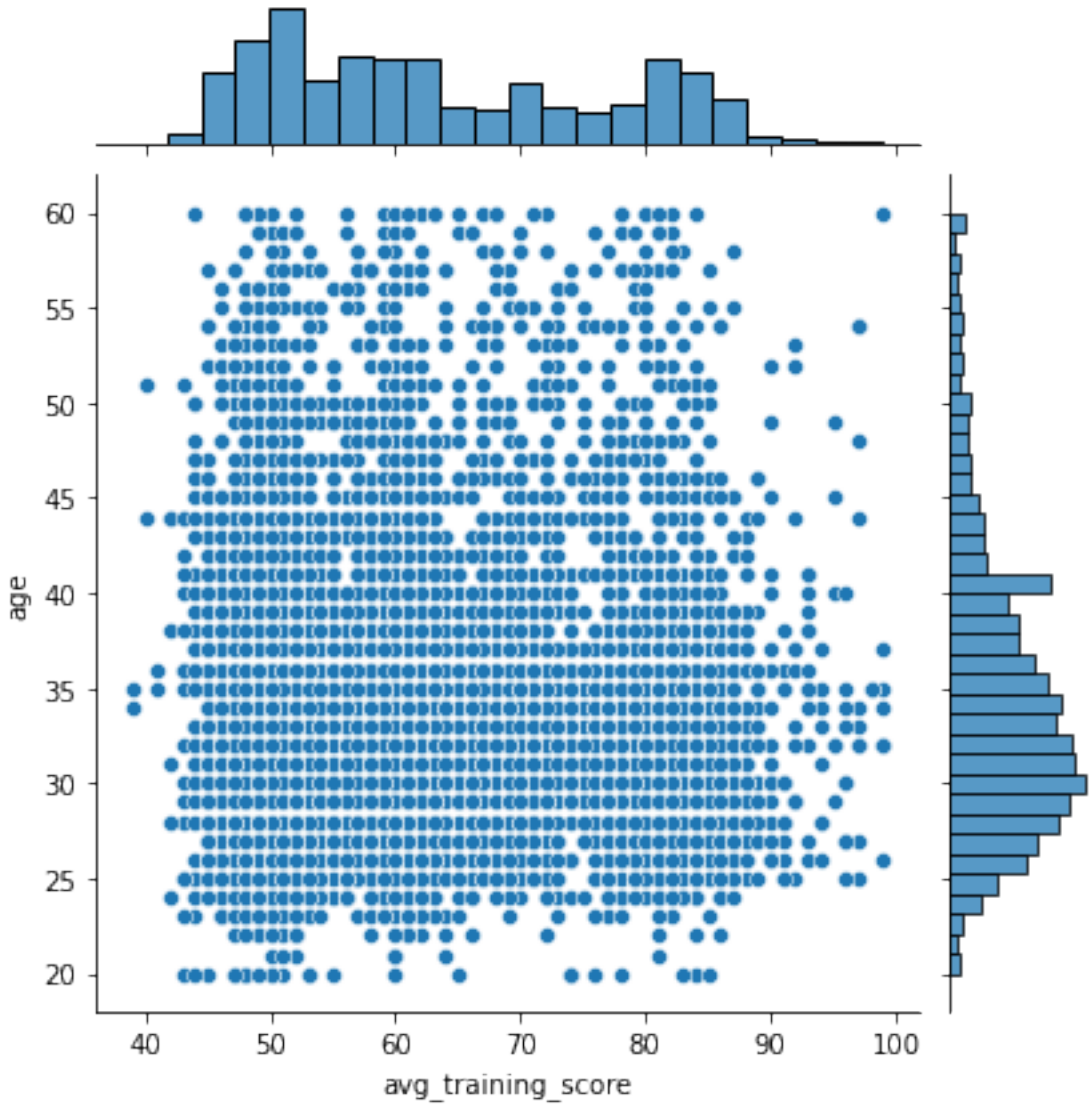


0.0.21 Plot Bivariate Distributions

0.0.22 Joint Plot

We can see how two independent variables are distributed with respect to each other ### Draw a joint plot between avg_training_score and age

[38]:
```
sns.jointplot(x="avg_training_score", y="age", data=df2);
```

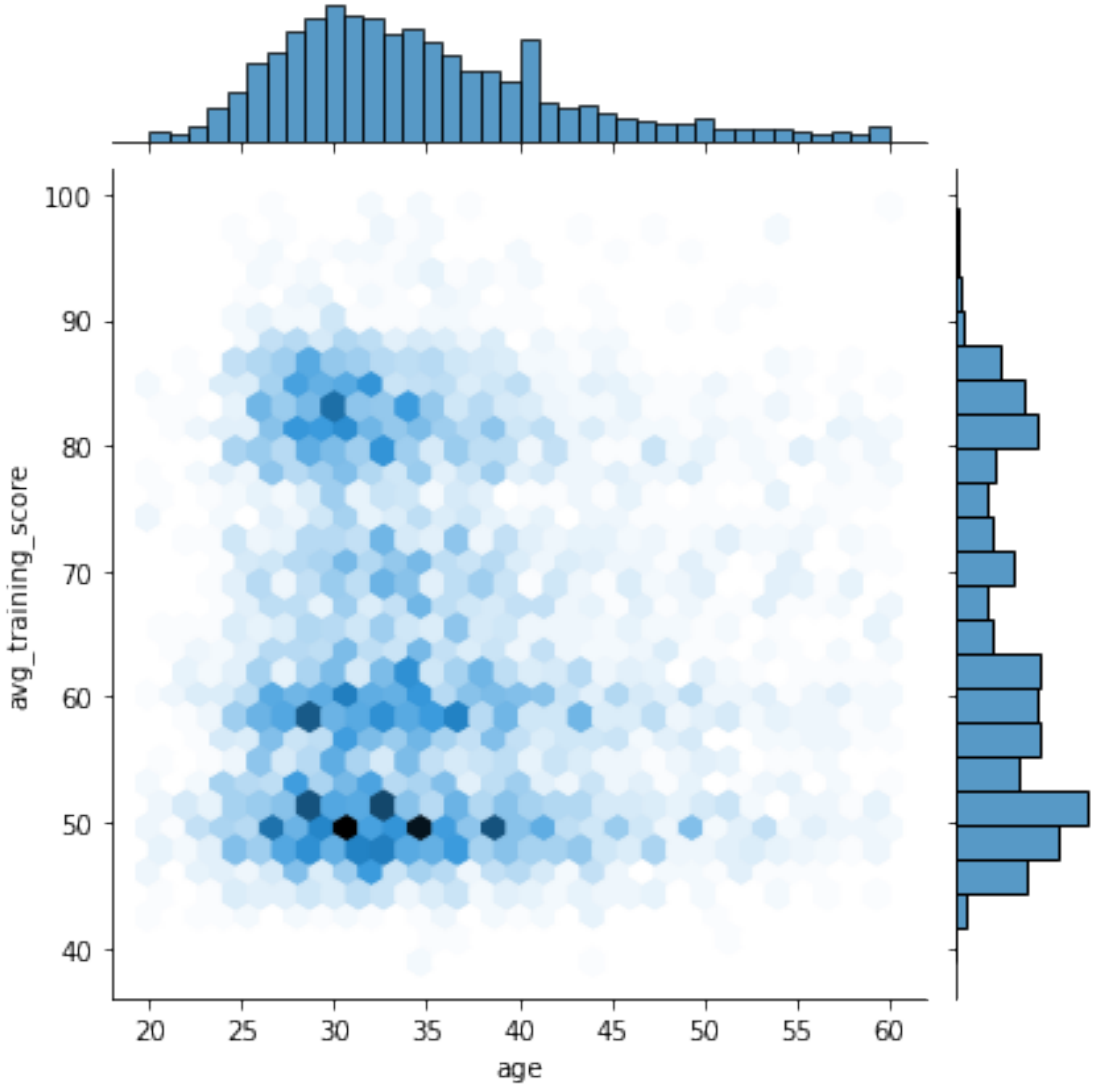


0.0.23 Hex Plot

Hexplot is a bivariate analog of histogram as it shows the number of observations that falls within hexagonal bins. Hexagonal binning is used in bivariate data analysis when the data is sparse in density i.e., when the data is very scattered and difficult to analyze through scatterplots ### Draw a hexplot for depicting the relationship between avg training score and age

[39]:
```
sns.jointplot(x=df2.age, y=df2.avg_training_score, kind="hex", data = df2)
```

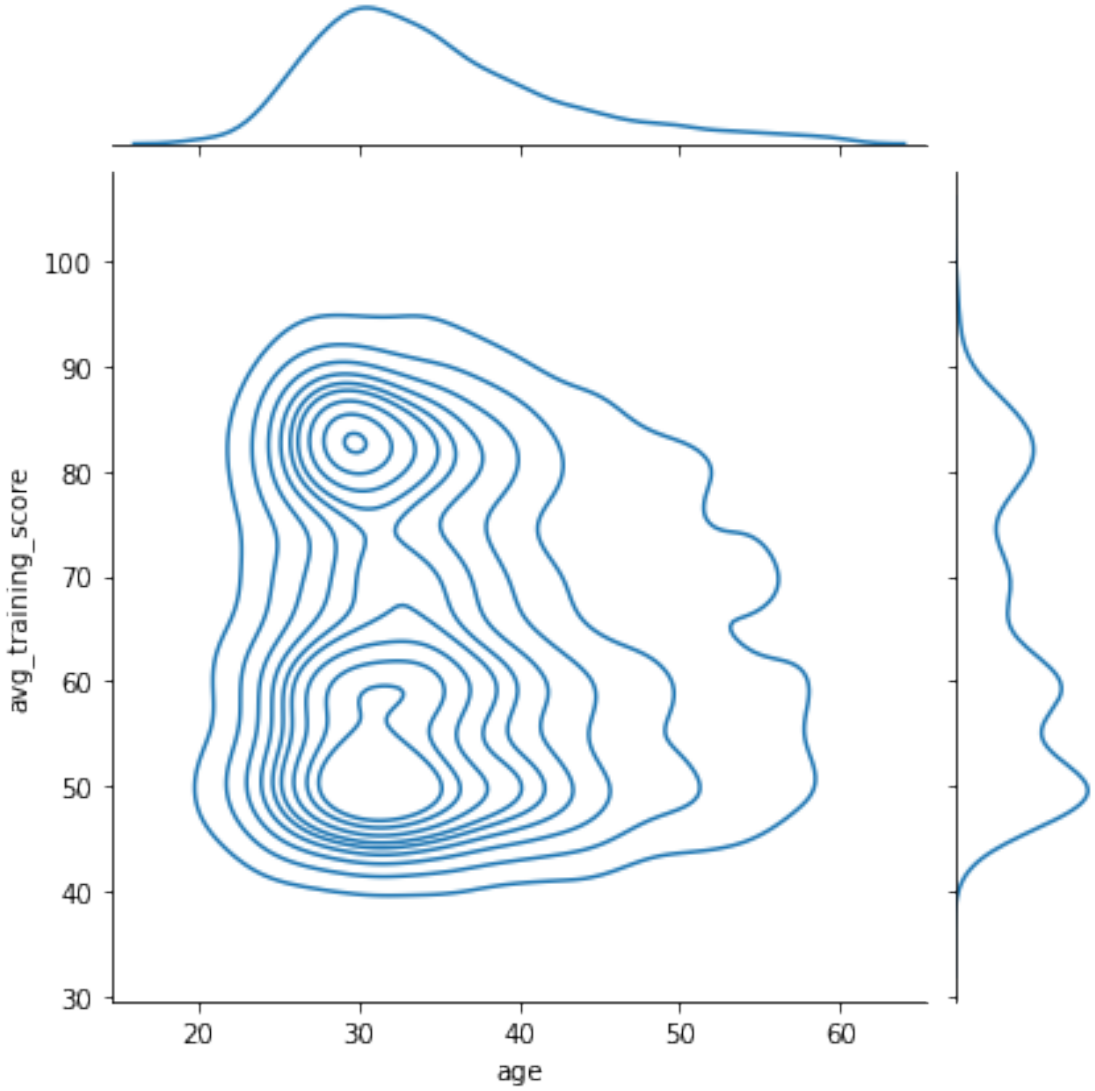
[39]:
```
<seaborn.axisgrid.JointGrid at 0x1e10955fc40>
```



0.0.24 KDE Plot

It is also possible to use the kernel density estimation procedure to visualize a bivariate distribution. In seaborn, this kind of plot is shown with a contour plot and is available as a style in jointplot()to visualize the bivariate distribution. ### Show KDE Plot to visualize age vs avg training score

[40]:
```
sns.jointplot(x="age", y="avg_training_score", data=df2, kind="kde");
```

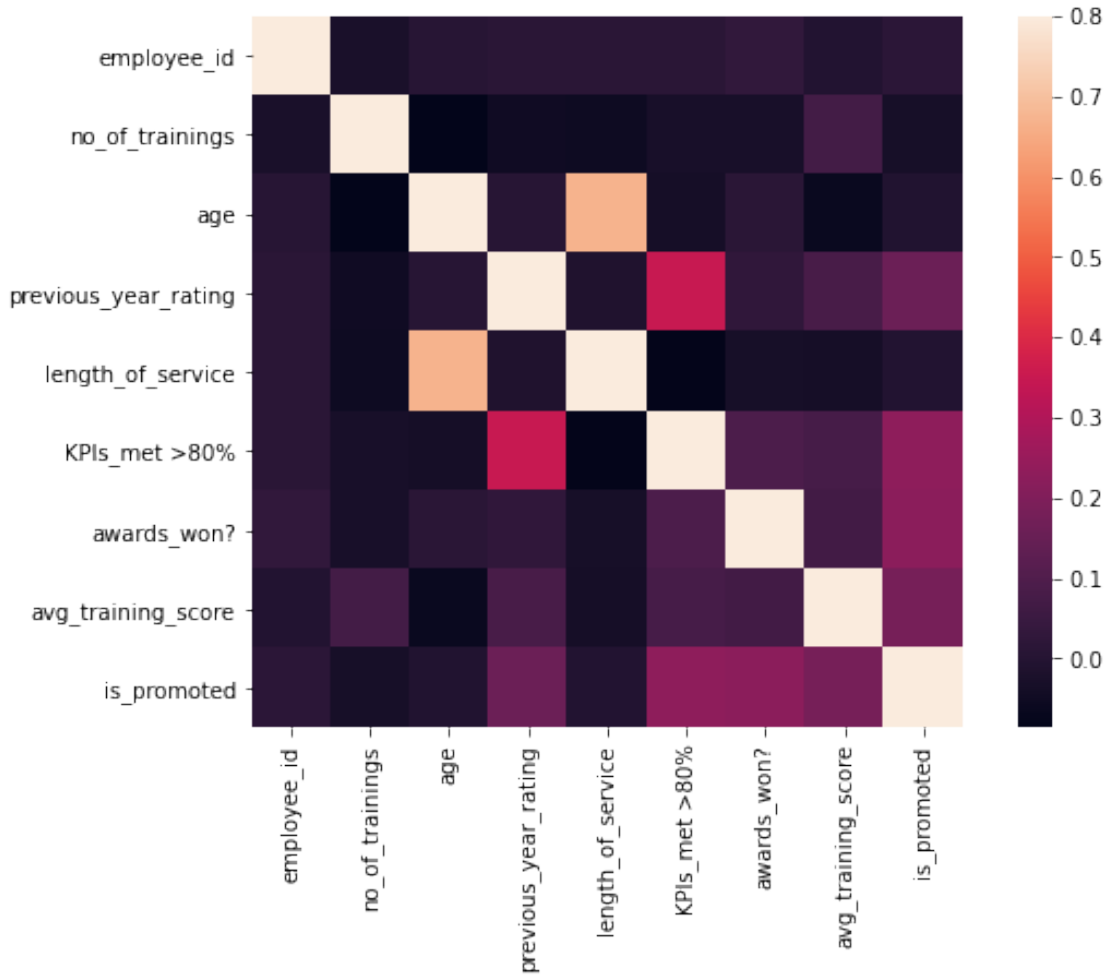


0.0.25 Heat Map

If you have a dataset with many columns, a good way to quickly check correlations among columns is by visualizing the correlation matrix as a heatmap.The stronger the color, the larger the correlation magnitude between columns. ### Draw heatmap for the dataset

[41]:
```
corrmat = df2.corr()
f, ax = plt.subplots(figsize=(9, 6))
sns.heatmap(corrmat, vmax=.8, square=True)
```

[41]: <AxesSubplot:>

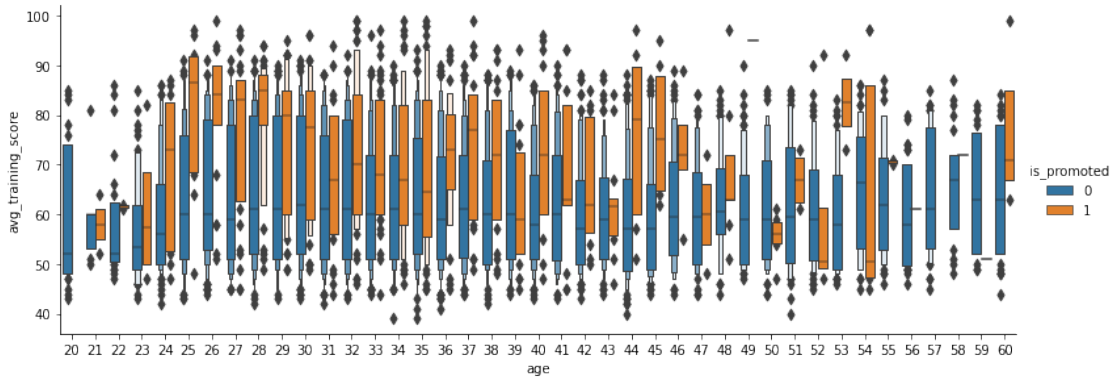


0.0.26 Boxen Plot

Boxen plots is used to to show the bivariate distribution. It shows large number of values of a variable, also known as quantiles. These quantiles are also defined as letter values. By plotting a large number of quantiles, it provides more insights about the shape of the distribution. ### Draw Boxen Plot between "age" and "avg_training_score, with hue"is_promoted" ### Adjust height and aspect values to make chart pretty

[42]:
```
sns.catplot(x="age", y="avg_training_score", data=df2, kind="boxen",height=4,␣
↪aspect=2.7, hue = "is_promoted")
```

[42]:
```
<seaborn.axisgrid.FacetGrid at 0x1e109341c40>
```

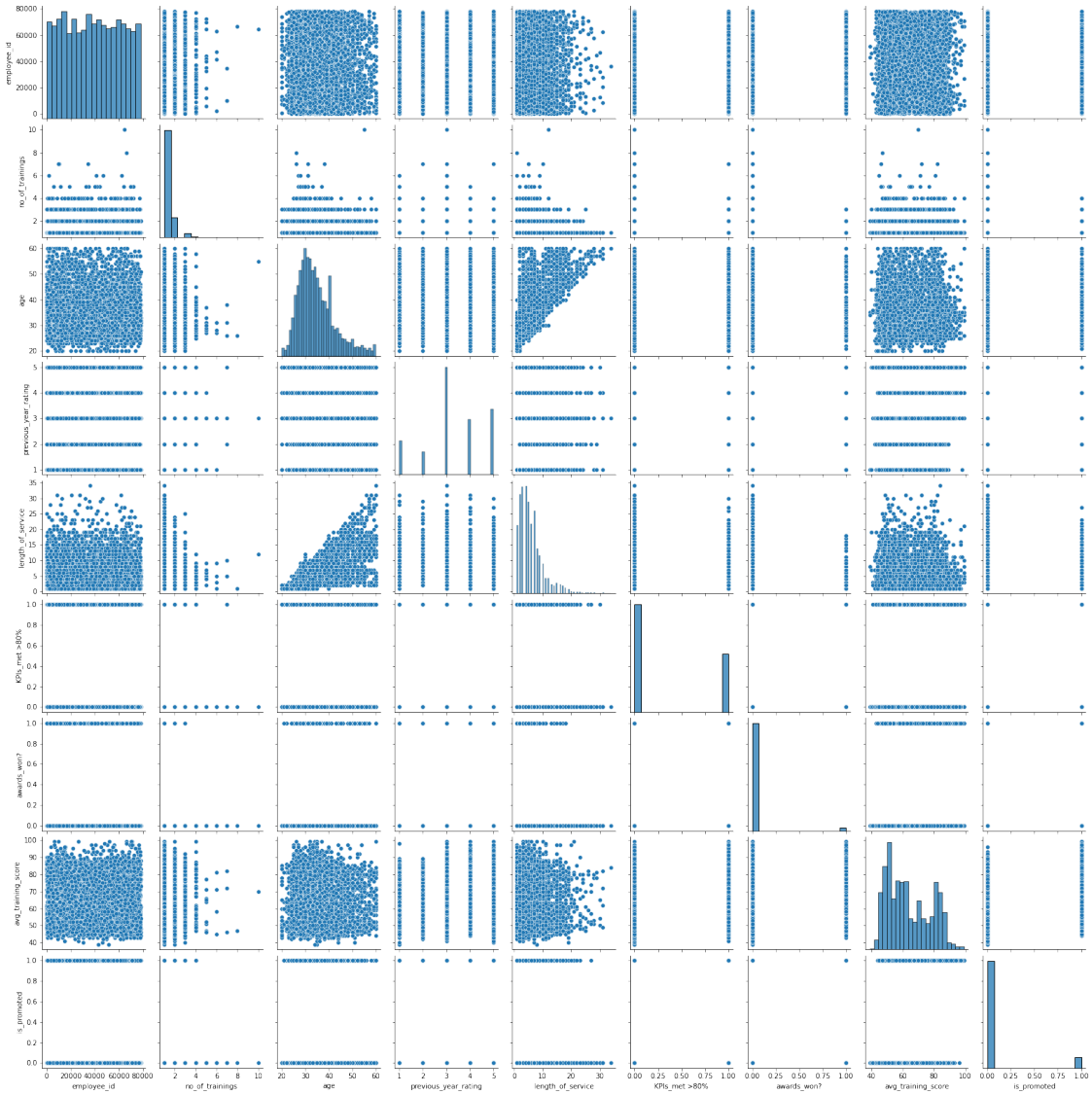


0.0.27 Pair Plot

We can also plot multiple bivariate distributions in a dataset by using pairplot() function of the seaborn library. This shows the relationship between each column of the database. It also draws the univariate distribution plot of each variable on the diagonal axis ### Draw a Pair Plot for the dataset

[43]:
```
sns.pairplot(df2)
```

[43]:
```
<seaborn.axisgrid.PairGrid at 0x1e10941ca00>
```


employee_id
no_of_trainings
age
previous_year_rating
length_of_service
KPIs_met >80%
awards_won?
avg_training_score
is_promoted