# Roll No: 205229133

## Lab7. Data Visualization in Seaborn

In [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
```

## Import train_upvote_mini.csv file

In [2]:

```
df = pd.read_csv("train_upvote_mini.csv")
df.head()
```

Out[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 |

## What is its size?

In [3]:

```
df.shape
```

Out[3]:

```
(15440, 7)
```

## Show the types of each feature

In [4]:

```
df.dtypes
```

Out[4]:

```
ID              int64
Tag            object
Reputation    float64
Answers       float64
Username        int64
Views         float64
Upvotes       float64
dtype: object
```

## How many unique "tag" available?

In [5]:

```
df.Tag.nunique()
```

Out[5]:

```
10
```

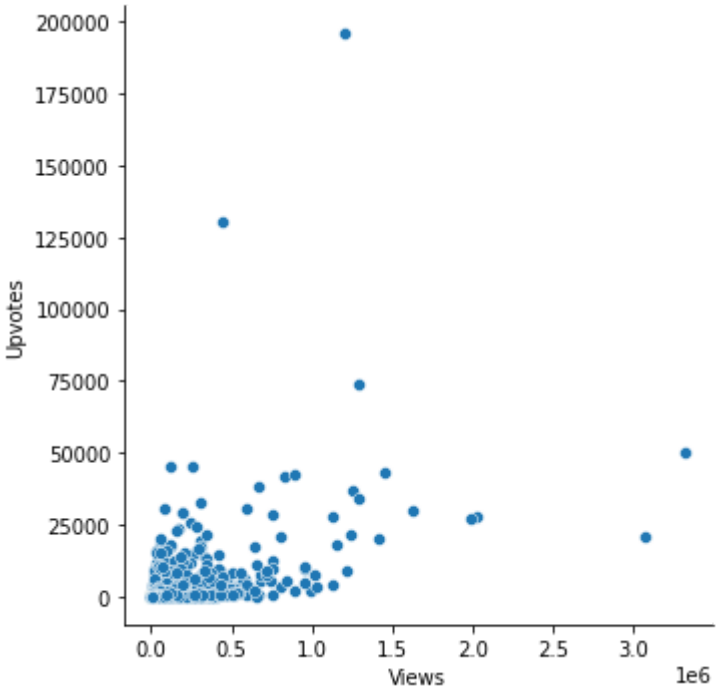
## Show scatterplot (inherited from matplotlib) and relplot between "views" and "upvotes"

In [6]:

```
sns.relplot(x="Views", y="Upvotes", data = df)
```

Out[6]:

```
<seaborn.axisgrid.FacetGrid at 0x243ae708160>
```



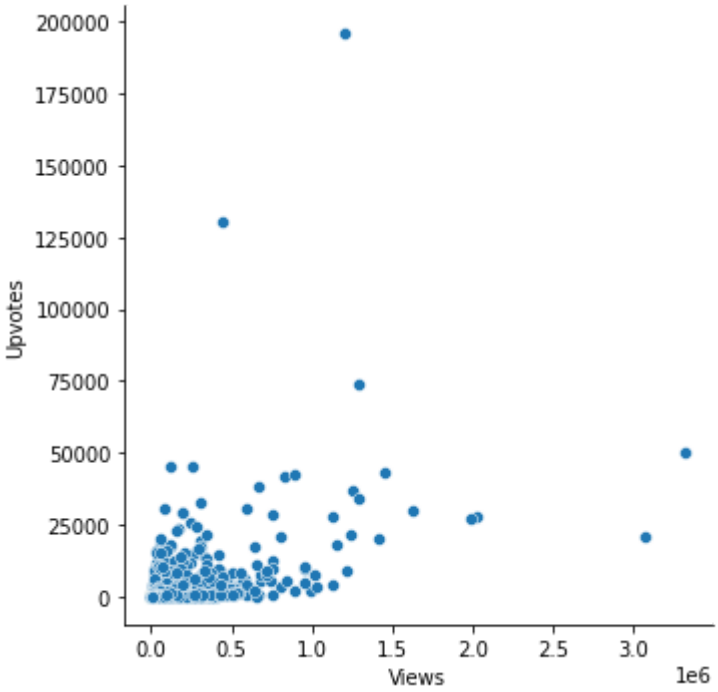

## Plot replot between "Views" and "Upvotes"

In [7]:

```
sns.relplot(x="Views", y="Upvotes", data = df)
```

Out[7]:

```
<seaborn.axisgrid.FacetGrid at 0x243b243e0d0>
```




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

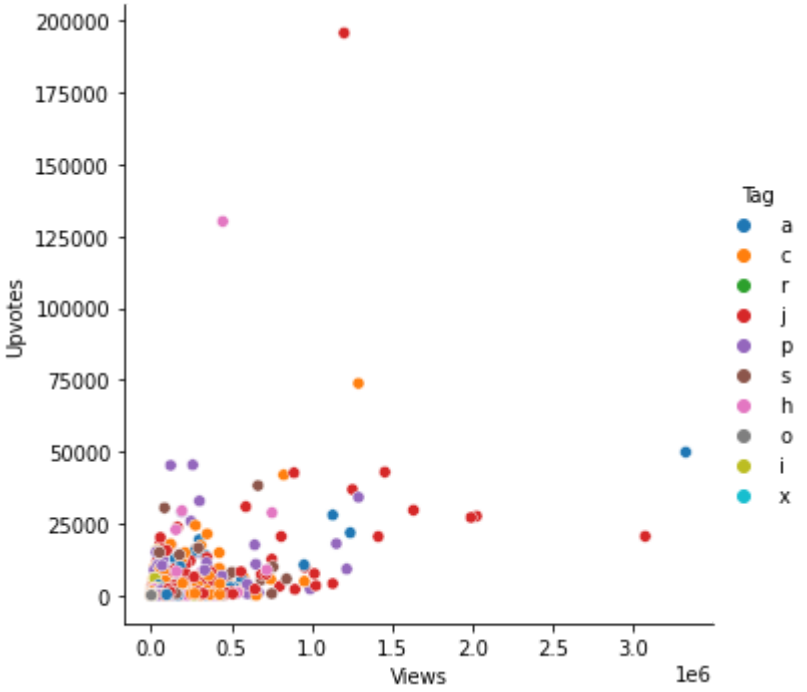
## Plot relplot between "Views" and "Upvotes" with hue as "Tag"

In [8]:

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

Out[8]:

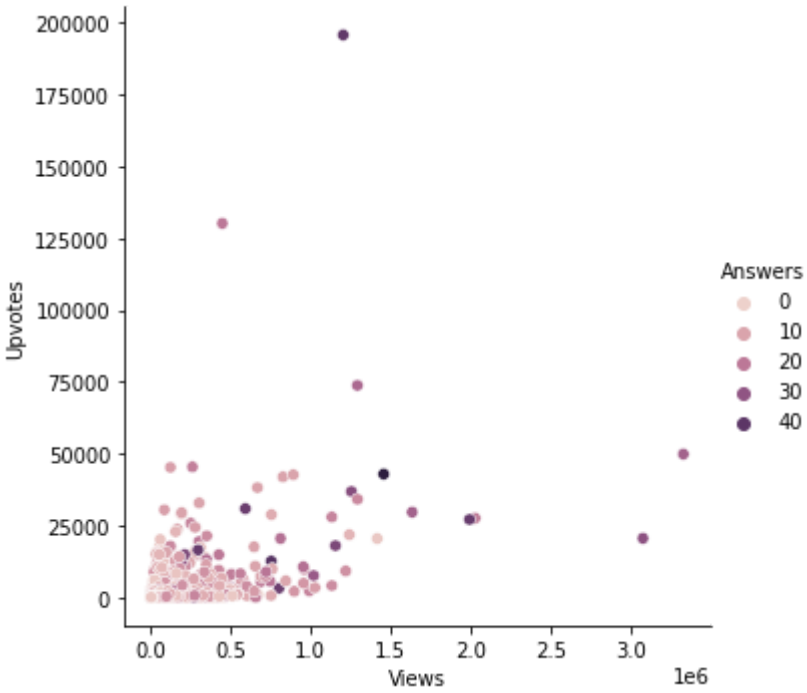
```
<seaborn.axisgrid.FacetGrid at 0x243b3bbb400>
```




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

In [9]:

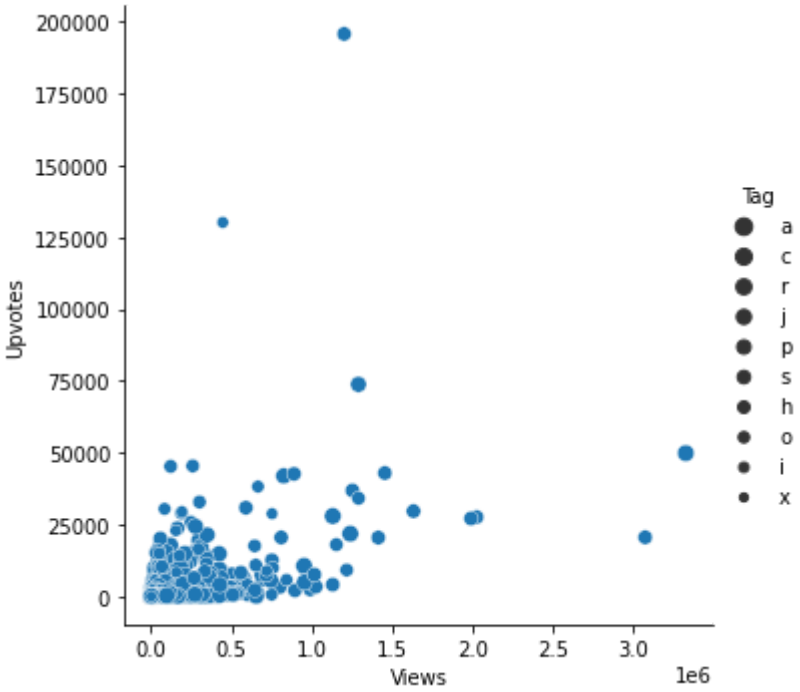
```
sns.relplot(x="Views", y="Upvotes", hue = "Answers", data = df);
```




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

In [10]:

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

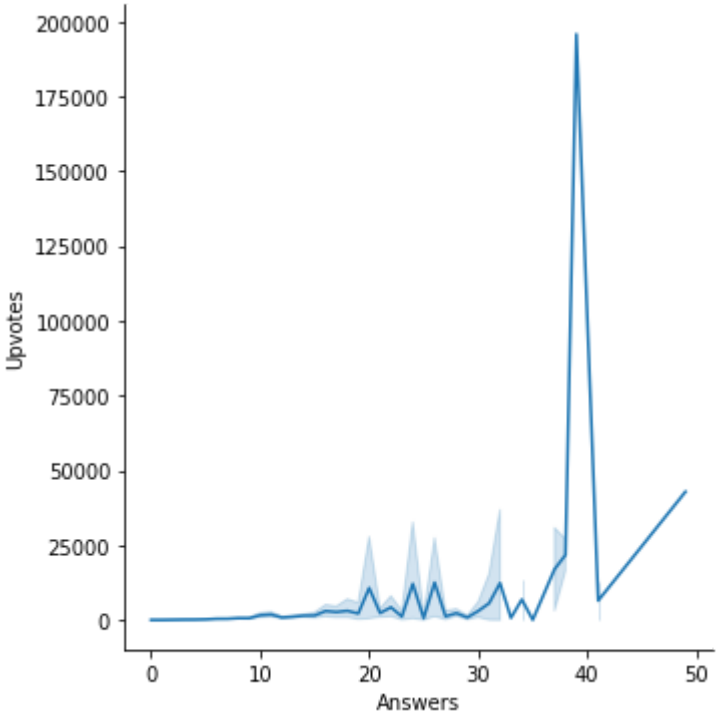



**Does no of times question answered impact the no. of upvotes?**

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

In [11]:

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

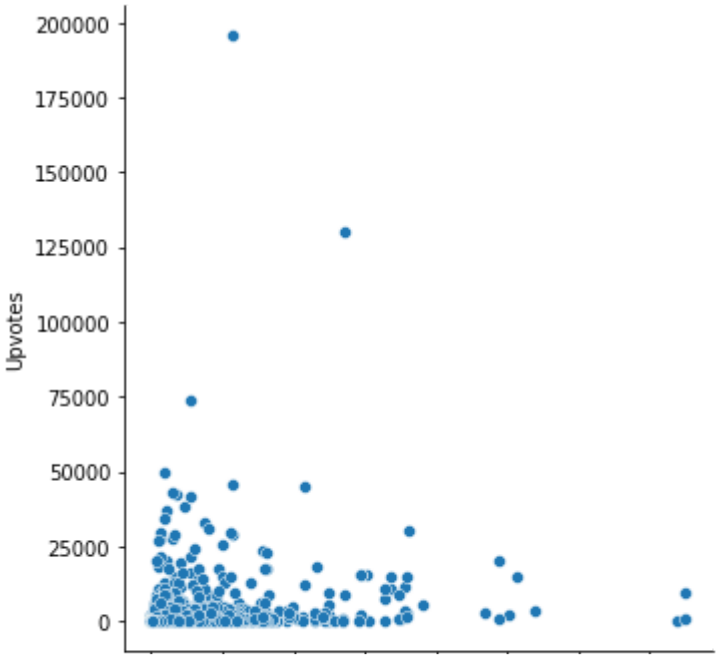



## Does Reputation score of question author impact no of upvotes?. Draw

## replot.

In [12]:

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




## Jitter Plot

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

In [13]:

```
df2 = pd.read_csv("train_hr_mini.csv")
df2.head()
```

Out[13]:

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

In [14]:

```
df2.shape
```

Out[14]:

```
(6397, 14)
```

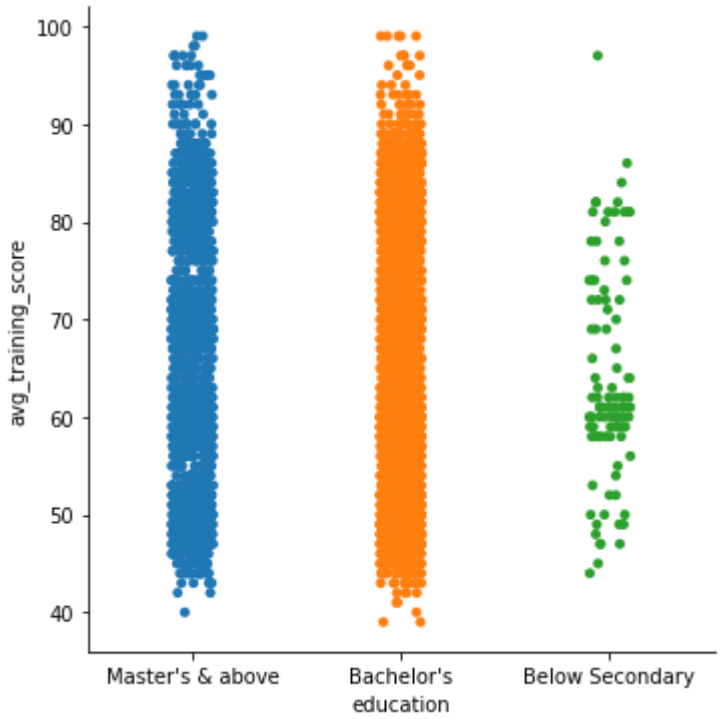
## Show Jitter plot between education and avg_training_score

In [15]:

```
sns.catplot(x="education", y="avg_training_score", data=df2)
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x243b3ebb910>
```




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.

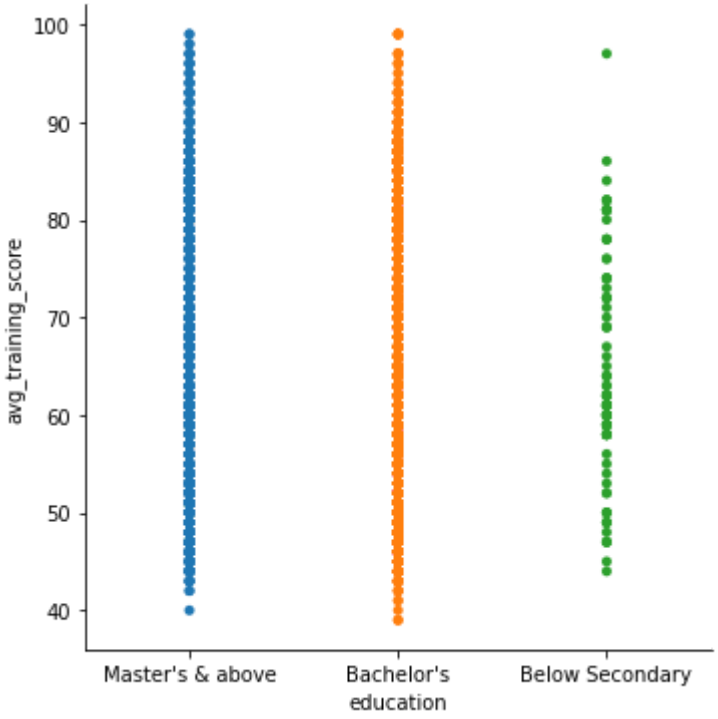
## Show the Jitter Plot

In [16]:

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

Out[16]:

```
<seaborn.axisgrid.FacetGrid at 0x243b3e46dc0>
```




## 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

In [17]:

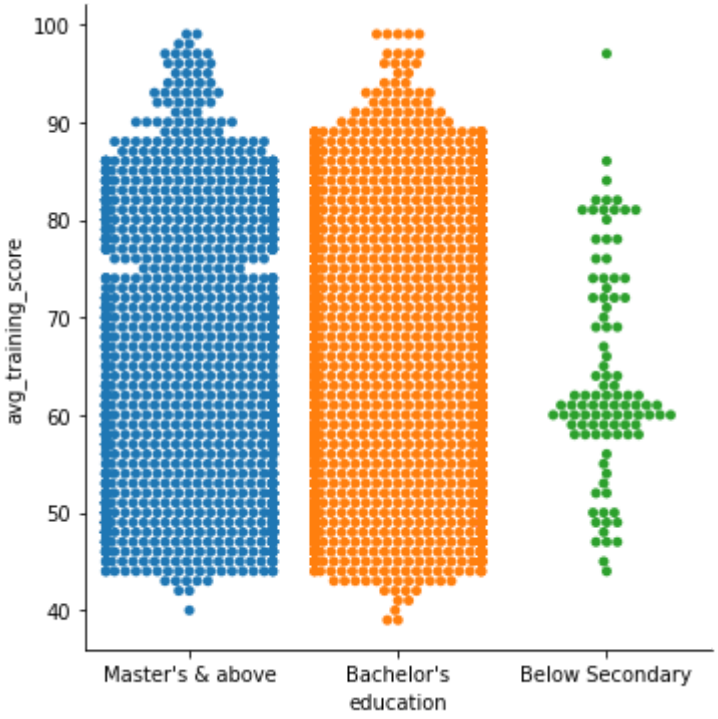
```
import warnings
warnings.filterwarnings('ignore')
```

In [18]:

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

Out[18]:

```
<seaborn.axisgrid.FacetGrid at 0x243b51171c0>
```




## 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

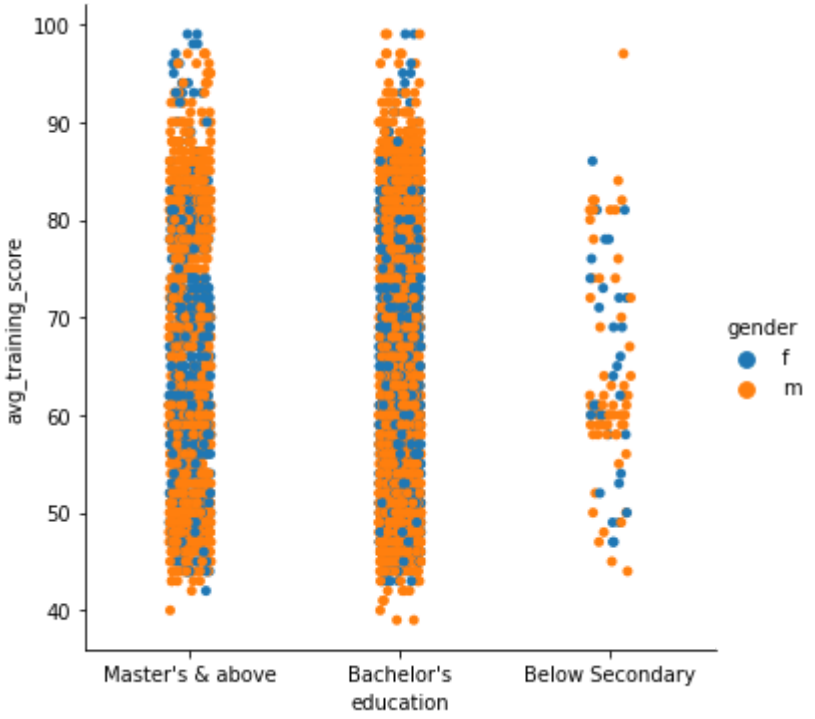
## Show Hue Plot to see the gender distribution in the plot of education category and avg_training_score.Here, hue is "gender".

In [19]:

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

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x243b5008940>
```



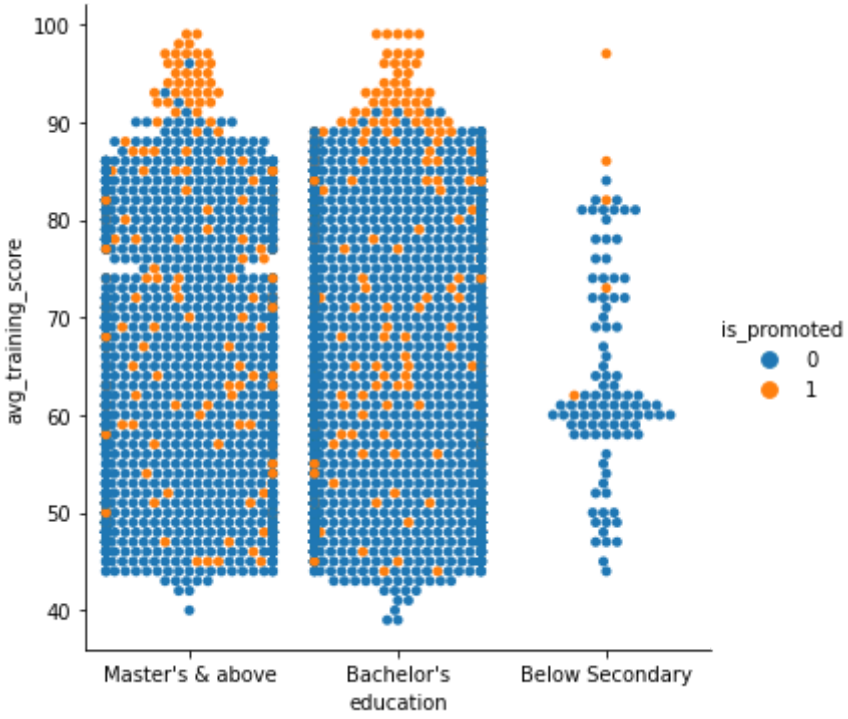

## Who are all promoted considering education and avg training score?. Draw swarm plot with hue as "is_promoted"

In [20]:

```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind = "swarm", dat
```

Out[20]:

```
<seaborn.axisgrid.FacetGrid at 0x243b3cdbe50>
```




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.

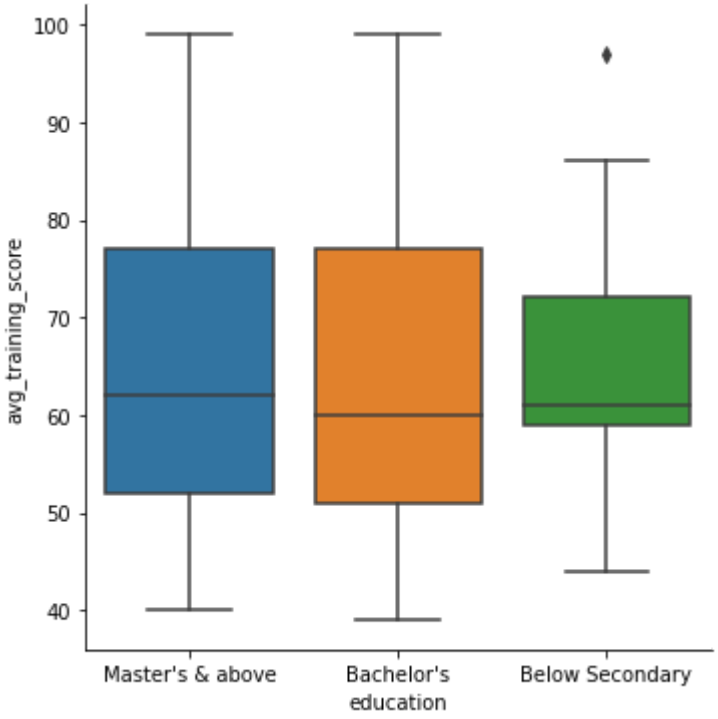
## Draw box plot between education and avg_training_score

In [21]:

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

Out[21]:

```
<seaborn.axisgrid.FacetGrid at 0x243b3ce62b0>
```




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

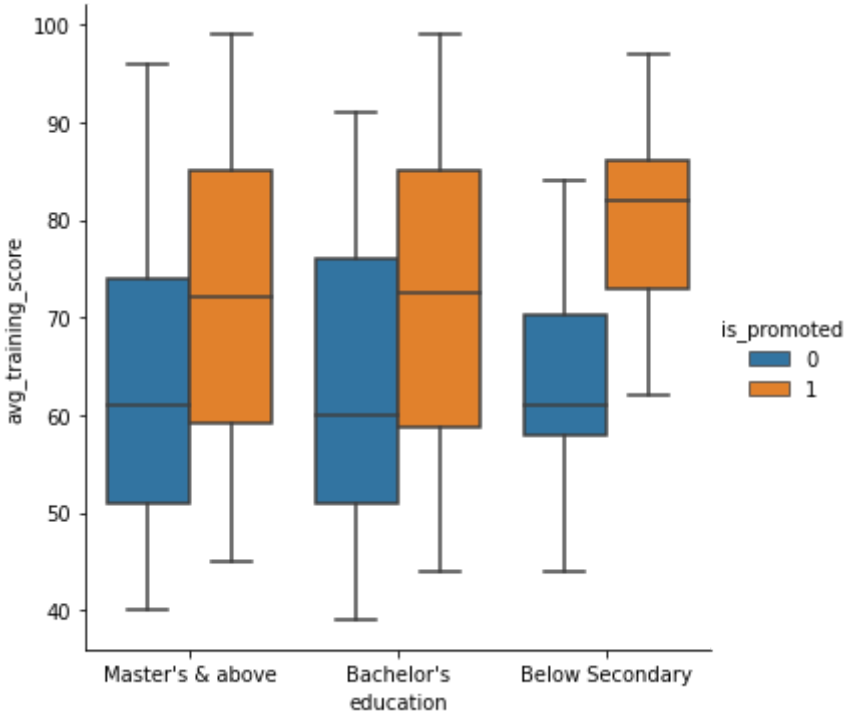
## Who are promoted and not promoted considering education and avg_training_score?. Draw Box Plot.

In [22]:

```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind = "box", data=
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x243b3de3430>
```




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.

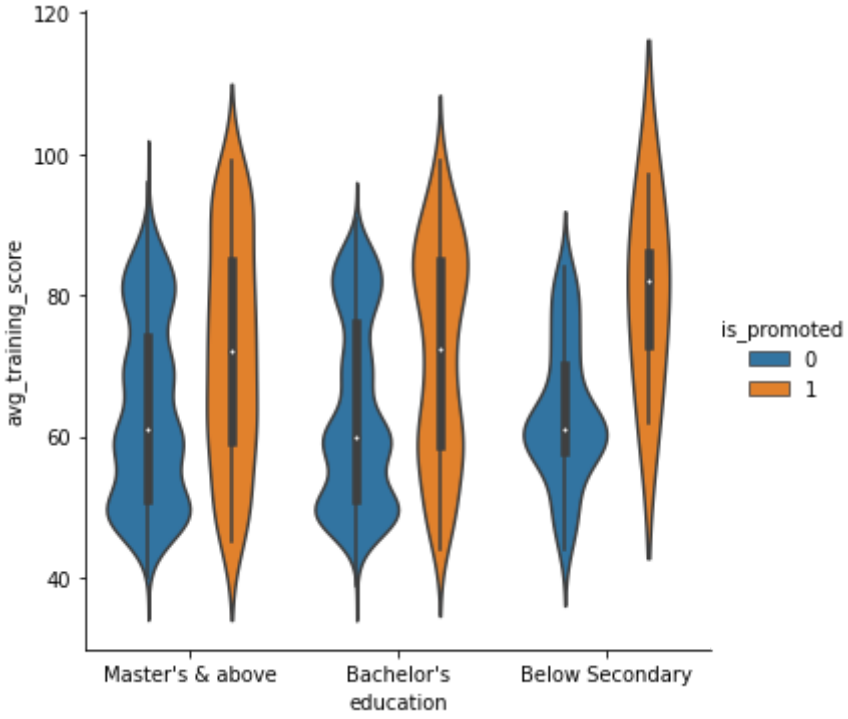
## Show violin plot between education categories and avg training score with hue as "is_promoted" target variable

In [23]:

```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind = "violin", da
```

Out[23]:

```
<seaborn.axisgrid.FacetGrid at 0x243b4f44370>
```




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.

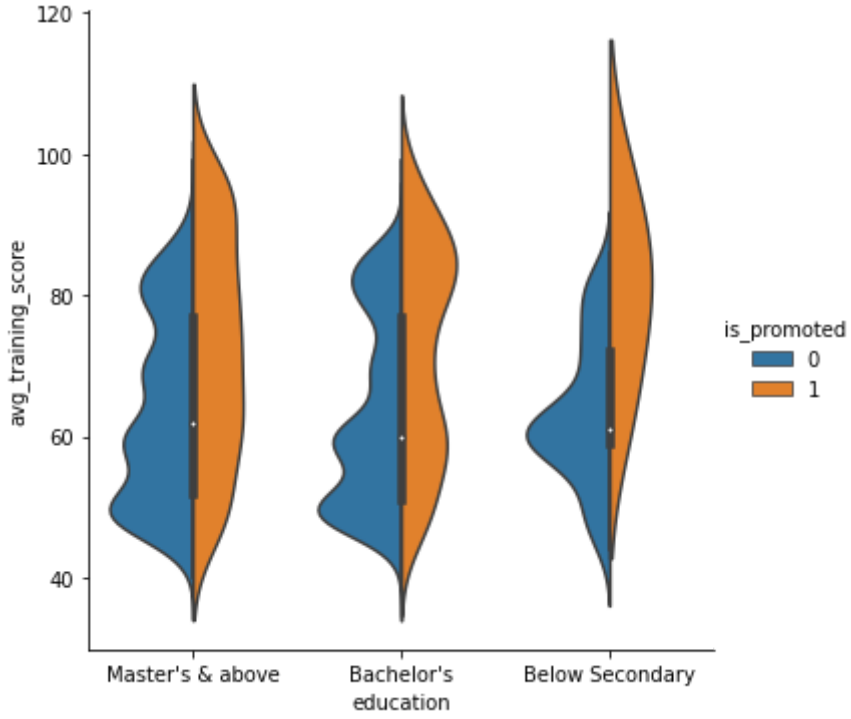
## Draw Violin plot with only 2 hue levels, use split attribute

In [24]:

```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind = "violin", sp
```

Out[24]:

```
<seaborn.axisgrid.FacetGrid at 0x243b4fb2460>
```



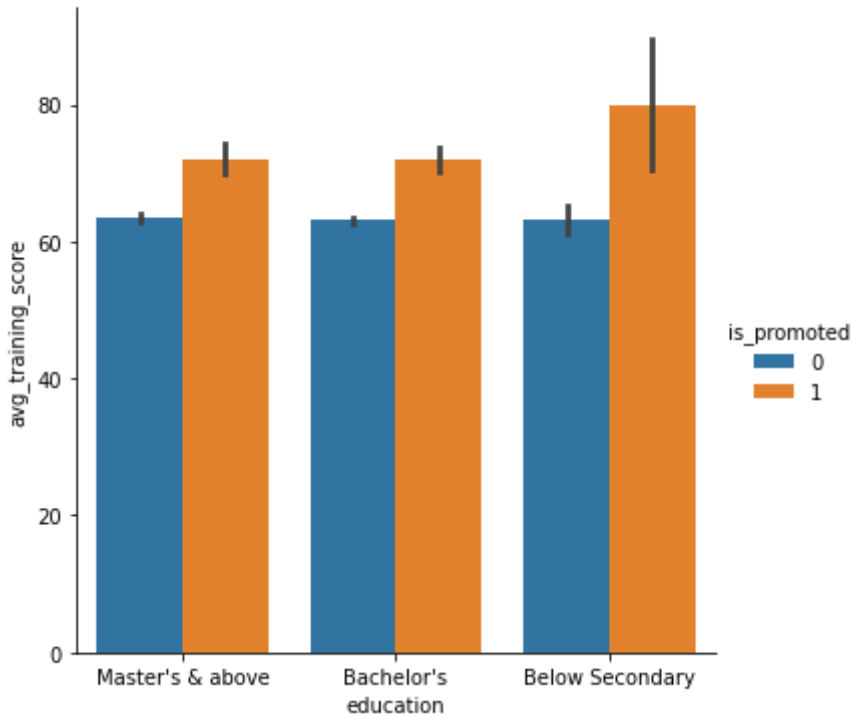

## Using catplot(), draw a Bar Chart between "education" and "avg_training_score", with hue as "is_promoted"

In [25]:

```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind = "bar", data=
```

Out[25]:

```
<seaborn.axisgrid.FacetGrid at 0x243b61a5730>
```




## 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.

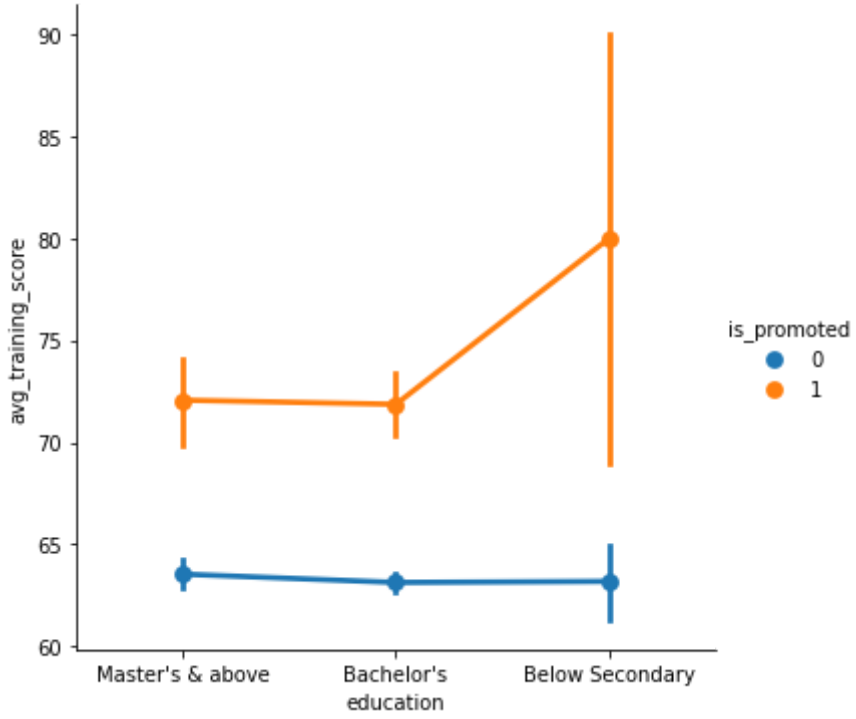
## Show point plot between education and avg training score with hue promotion category

In [26]:

```
sns.catplot(x="education", y="avg_training_score", hue = "is_promoted", kind = "point", dat
```

Out[26]:

```
<seaborn.axisgrid.FacetGrid at 0x243b61923d0>
```




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.

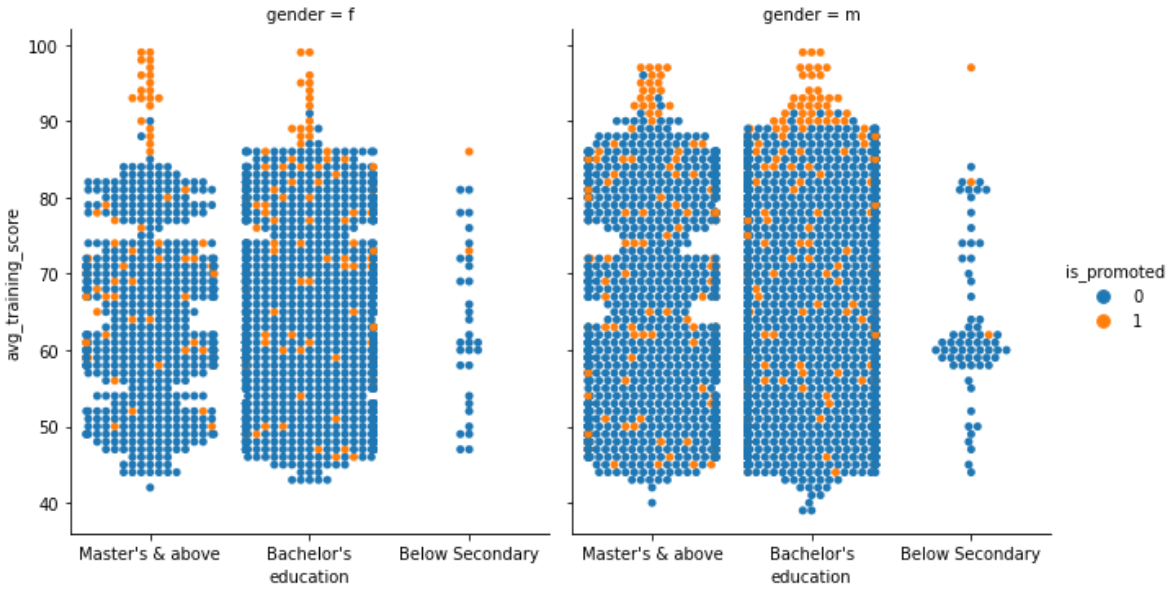
## Multiple Dimension in Seaborn

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

In [27]:

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




## Plot Univariate Distributions

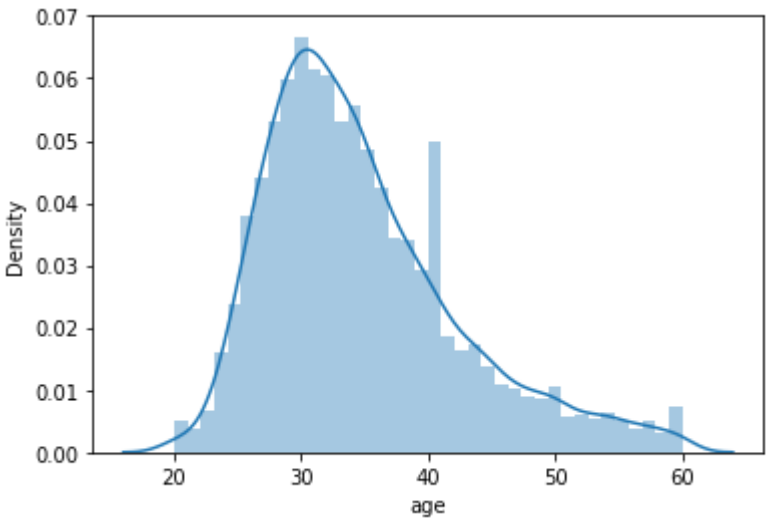
### Plot Histogram with kernel density estimate value for age attribute

In [28]:

```
sns.distplot(df2.age)
```

Out[28]:

```
<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.

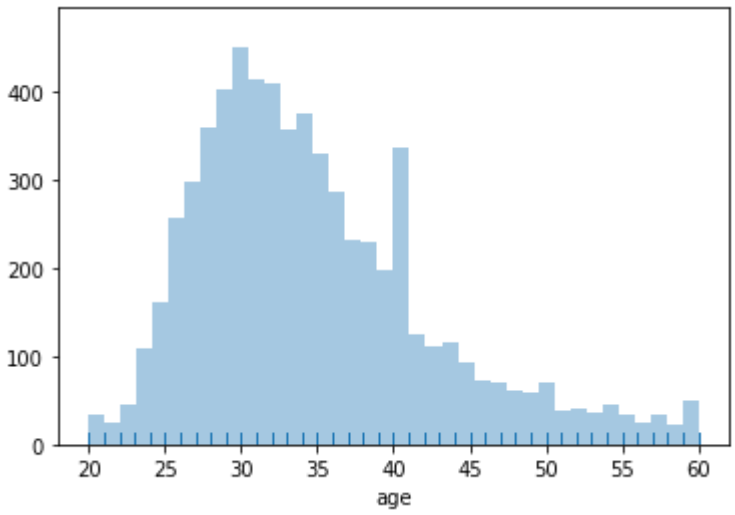
## Show only Histogram for age variable, without KDE

In [29]:

```
sns.distplot(df2.age, kde=False, rug = True)
```

Out[29]:

```
<AxesSubplot:xlabel='age'>
```




## Plot Bivariate Distributions

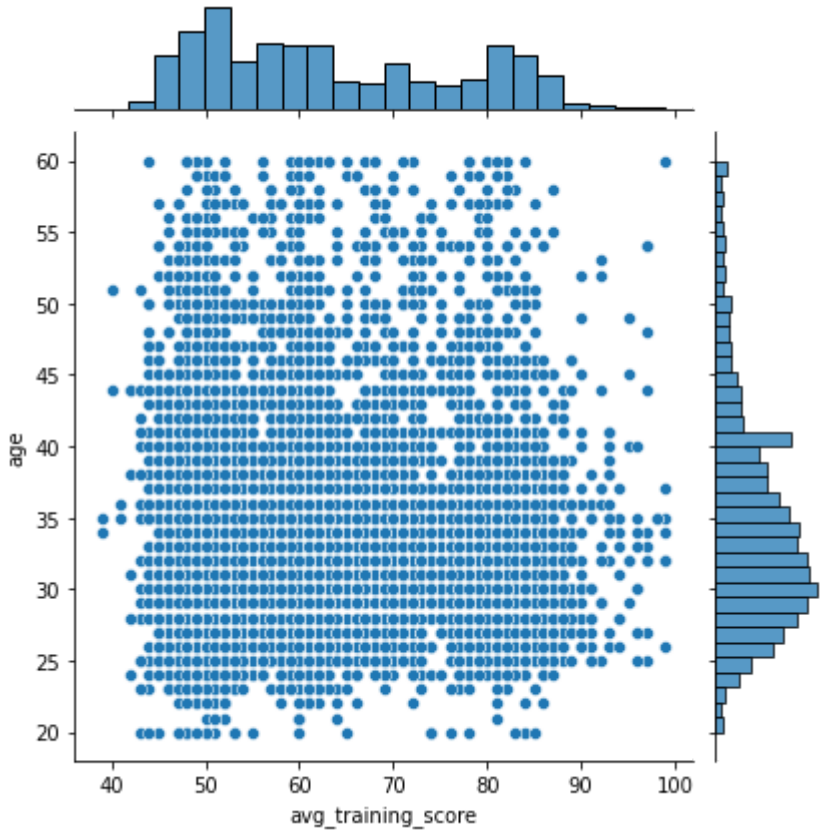
## Joint Plot

We can see how two independent variables are distributed with respect to each other

## Draw a joint plot between avg_training_score and age

In [30]:

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




## 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

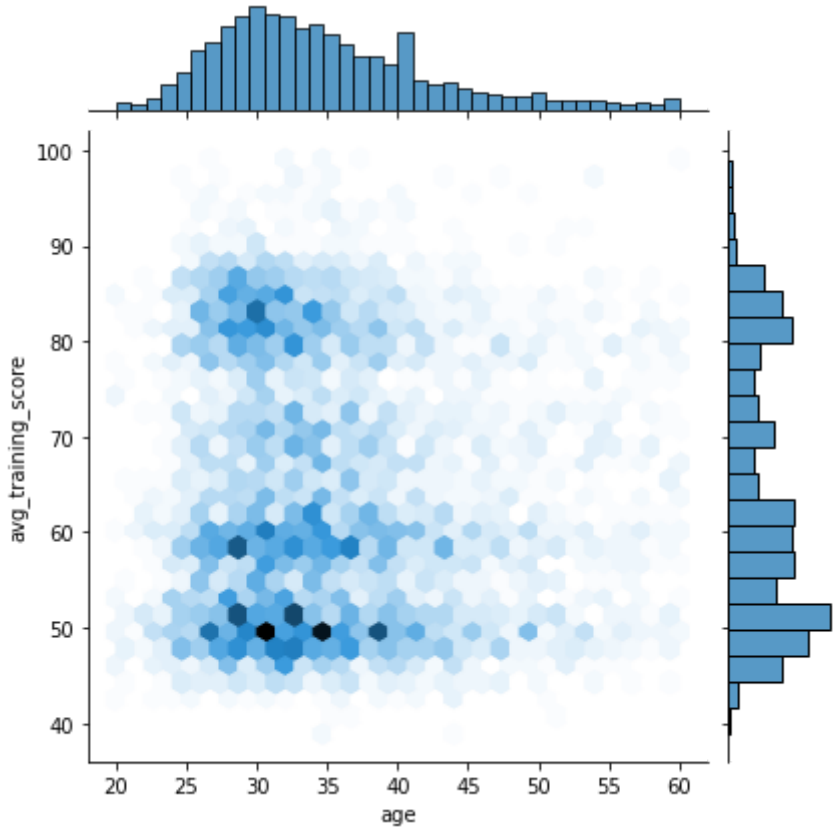
## Draw a hexplot for depicting the relationship between avg training score and age

In [31]:

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

Out[31]:

```
<seaborn.axisgrid.JointGrid at 0x243b689a910>
```



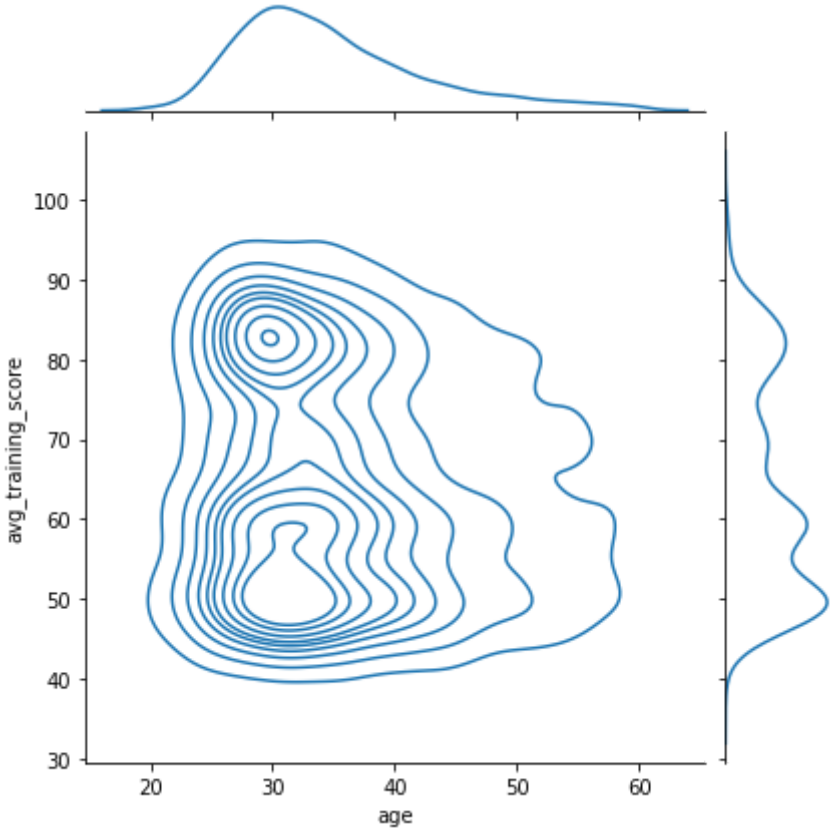

## KDE Plot

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

In [32]:

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




## 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.

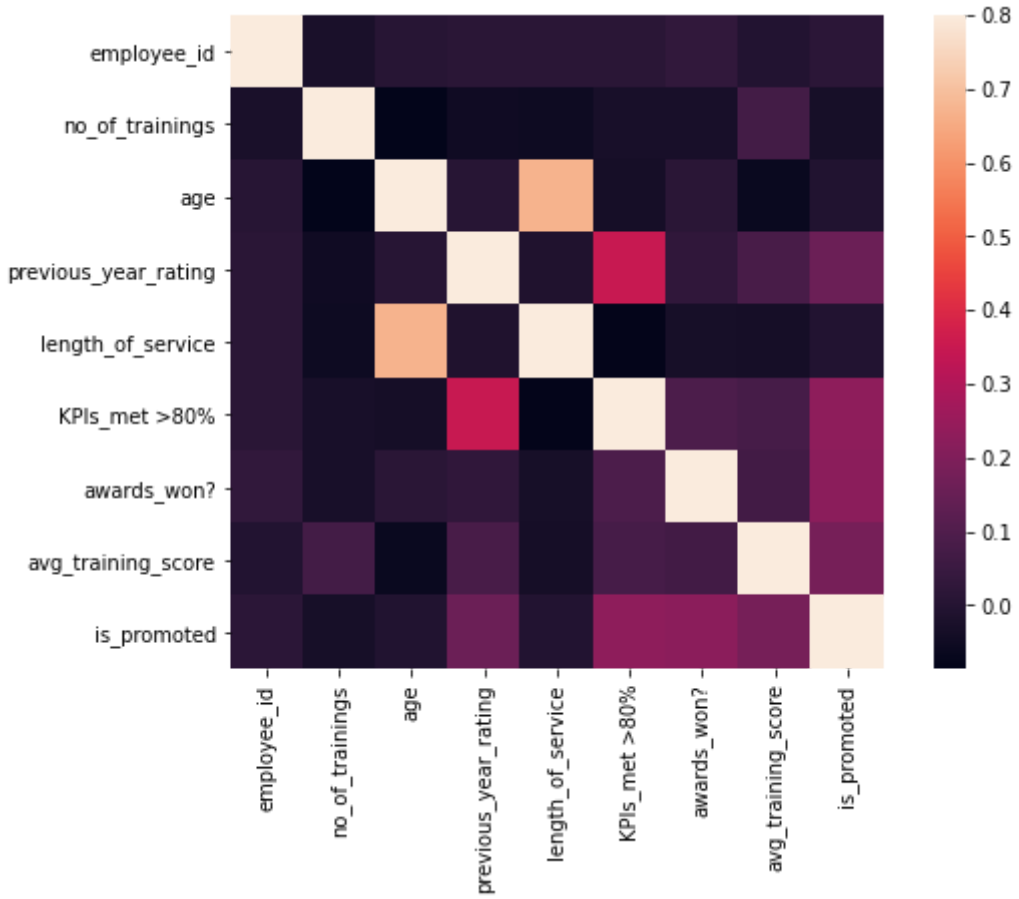
## Draw heatmap for the dataset

In [33]:

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

Out[33]:

<AxesSubplot:>




## 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"

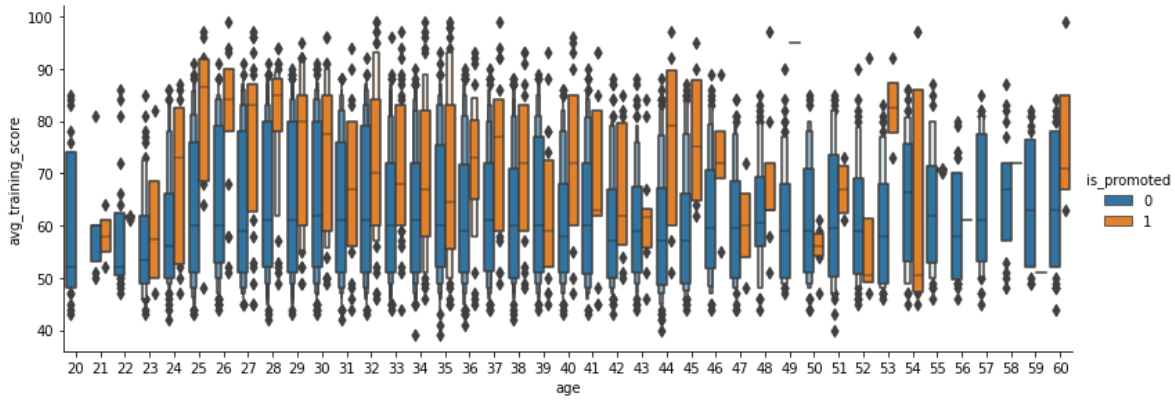
## Adjust height and aspect values to make chart pretty

In [34]:

```
sns.catplot(x="age", y="avg_training_score", data=df2, kind="boxen",height=4, aspect=2.7, h
```

Out[34]:

```
<seaborn.axisgrid.FacetGrid at 0x243b612ebe0>
```




## 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

In [35]:

```
sns.pairplot(df2)
```

Out[35]:

```
<seaborn.axisgrid.PairGrid at 0x243b6b50a30>
```