Department of Data Science - Data and Visual Analytics Lab

Lab7. Data Visualization in Seaborn

Objectives

After completing this lab, you will learn how to

- Visualize Statistical Relationships using Scatter plot, relplot, Hue plot, Line plot
- Plot Categorical Data using Jitter plot, swarm plot, violin plot, box plot, point plot
- Visualize the distribution of dataset using Histogram, Hexplot, KDE plot, Boxen plot
- perform Pairwise correlation using Heatmap
- Understand Multiple bivariate relationships using Pairplot

Dataset - Online Question and Answer Platform

An online question and answer platform has hired you as a data scientist to identify the best question authors on the platform. This identification will bring more insight into increasing the user engagement. The tag of the question, number of views received, number of answers, username and reputation of the question author are given in this dataset. The problem requires you to predict the upvote count that the question will receive.

Variable	Definition
ID	Question ID
Tag	Anonymised tags representing question category
Reputation	Reputation score of question author
Answers	Number of times question has been answered
Username	Anonymised user id of question author
Views	Number of times question has been viewed
Upvotes	(Target) Number of upvotes for the question

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
```

1. Visualizing Statistical Relationships

A statistical relationship denotes a process of understanding relationships between different variables in a dataset and how that relationship affects or depends on other variables.

Here, we'll be using seaborn to generate the below plots:

```
Scatter plot
relplot
Hue plot
Line plot
```

In this exercise, let us Predict the number of upvotes

Import train_upvote_mini.csv file

In [2]:
```
df = pd.read_csv("./visualization_data/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 [13]:
```
df.shape
```

Out[13]: (15440, 7)

Show the types of each feature

In [ ]:
```
df.dtypes
```

How many unique "tag" available?

In [ ]: df.tag.nunique()

Visualize with Scatterplot

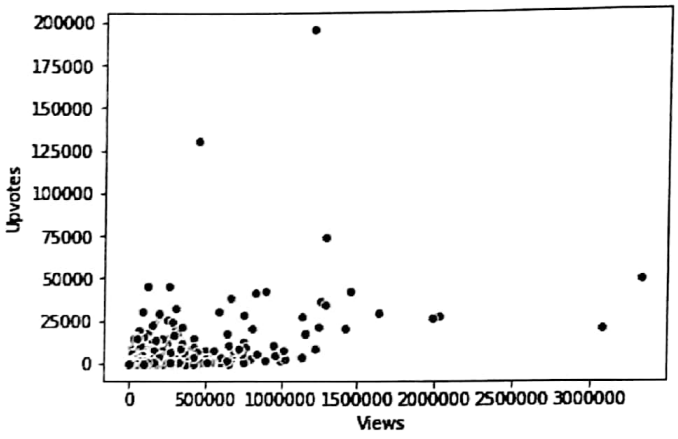
A scatterplot is perhaps the most common example of visualizing relationships between two variables. Each point shows an observation in the dataset and these observations are represented by dot-like structures. The plot shows the joint distribution of two variables using a cloud of points.

Does no. of views correlate no of upvotes?.

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

In [3]: sns.scatterplot(x="views", y="upvotes", data=df)

Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x1c9f75f00b8>



Plot replot between "Views" and "Upvotes"

Hue Plot

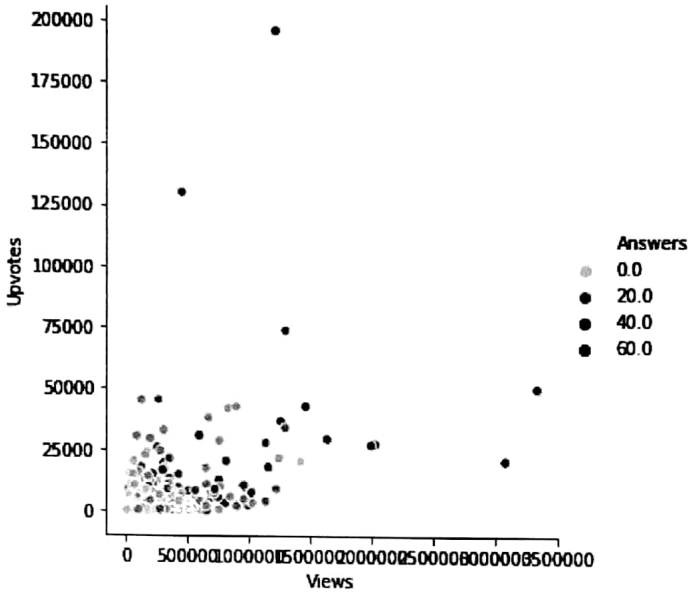
We can add another dimension in our plot with the help of hue as it gives color to the points and each color has some meaning attached to it.

In the above plot, the hue semantic is categorical. That's why it has a different color palette. If the hue semantic is numeric, then the coloring becomes sequential.

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

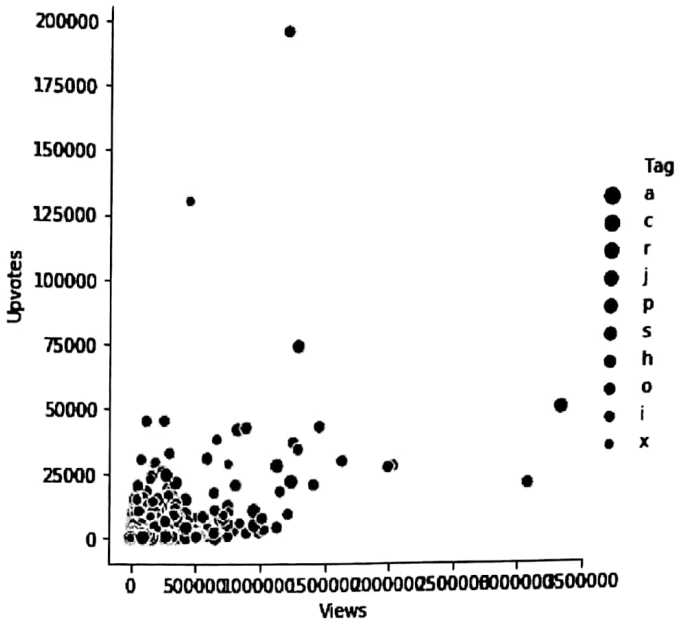
In [6]:
```
sns.relplot(x="Views", y="Upvotes", data=df)
```

Out[6]:
```
<seaborn.axisgrid.FacetGrid at 0x1c9f9772b70>
```



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

In [7]: sns.relplot(x="Views", y="Upvotes", hue="Tag", data=df)

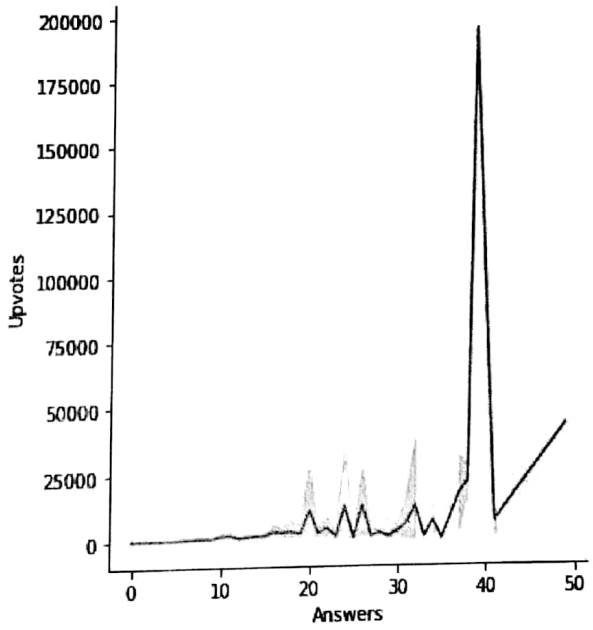


Does no of times question answered impact the no. of upvotes?

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

In [8]: sns.relplot(x="Answers", y="Upvotes", kind="line", data=df);

Out[8]: <seaborn.axisgrid.FacetGrid at 0x1c9f99a0c88>

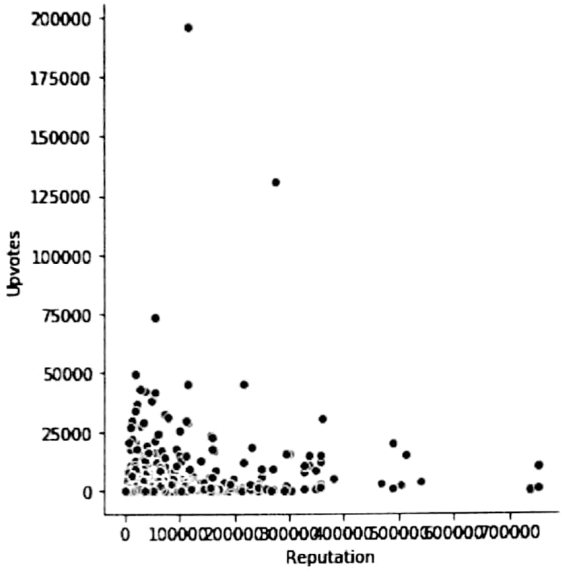


Does Reputation score of question author impact no of upvotes?. Draw replot.

In [9]: sns.relplot(data=df, x='Reputation', y='Upvotes')

Out[9]: <seaborn.axisgrid.FacetGrid at 0x1c9f9b03898>

plt.show()



2. Visualizating Categorial Data

Various Categorial Plots in Seaborn

Categorical scatterplots:

- `stripplot()` (with `kind="strip"`; the default)
- `swarmplot()` (with `kind="swarm"`)

Categorical distribution plots:

- `boxplot()` (with `kind="box"`)
- `violinplot()` (with `kind="violin"`)
- `boxenplot()` (with `kind="boxen"`)

Categorical estimate plots:

- `pointplot()` (with `kind="point"`)
- `barplot()` (with `kind="bar"`)
- `countplot()` (with `kind="count"`)

In the previous section, we saw how we can use different visual representations to show the relationship between multiple variables. We drew the plots between two numeric variables. In this section, we'll see the relationship between two variables of which one would be categorical (divided into different groups).

We'll be using catplot() function of seaborn library to draw the plots of categorical data using HR Analytics Dataset.

Dataset - HR analytics description

Your client is a large MNC and they have 9 broad verticals across the organisation. One of the problem your client is facing is around identifying the right people for promotion (only for manager position and below) and prepare them in time. Currently the process, they are following is:

1. They first identify a set of employees based on recommendations/ past performance
2. Selected employees go through the separate training and evaluation program for each vertical. These programs are based on the required skill of each vertical
3. At the end of the program, based on various factors such as training performance, KPI completion (only employees with KPIs completed greater than 60% are considered) etc., employee gets promotion

For above mentioned process, the final promotions are only announced after the evaluation and this leads to delay in transition to their new roles. Hence, company needs your help in identifying the eligible candidates at a particular checkpoint (ie., time frame from the time of nomination stage to a particular time point) so that they can expedite the entire promotion cycle.

They have provided multiple attributes around Employee's past and current performance along with demographics. Now, The task is to predict whether a potential promotee at checkpoint in the test set will be promoted or not after the evaluation process.

```
Features of HR analytics dataset
Variable    Definition
employee_id Unique ID for employee
department  Department of employee
region  Region of employment (unordered)
education   Education Level
gender  Gender of Employee
recruitment_channel Channel of recruitment for employee
no_of_trainings no of other trainings completed in previous year on soft skills, technical
skills etc.
age Age of Employee
previous_year_rating    Employee Rating for the previous year
length_of_service   Length of service in years
KPIs_met >80%   if Percent of KPIs(Key performance Indicators) >80% then 1 else 0
awards_won? if awards won during previous year then 1 else 0
avg_training_score  Average score in current training evaluations
is_promoted (Target) Recommended for promotion
```

Jitter Plot

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

In [11]:
```
df2 = pd.read_csv("./visualization_data/train_hr_mini.csv")
df2.head()
```

Out[11]:

	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

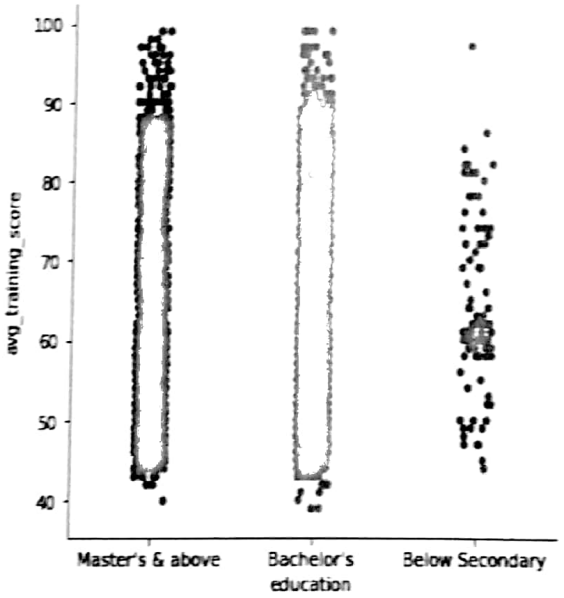
In [12]:
```
df2.shape
```

Out[12]:
```
(6397, 14)
```

Show Jitter plot between education and avg_training_score

In [14]:
```
sns.catplot(x="education", y="avg_training_score", data=df2)
```

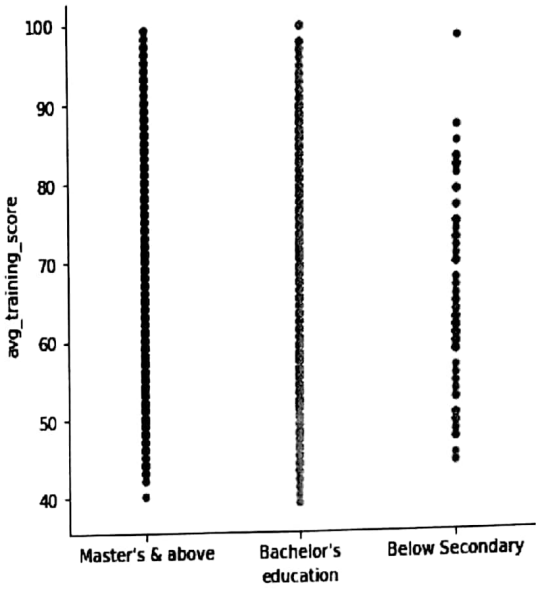
Out[14]:
```
<seaborn.axisgrid.FacetGrid at 0x1c9fadc67f0>
```



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.

In [15]: sns.catplot(x="education", y="avg_training_score", jitter=False, data=df)

Out[15]: <seaborn.axisgrid.FacetGrid at 0x1c9fae66cf8>



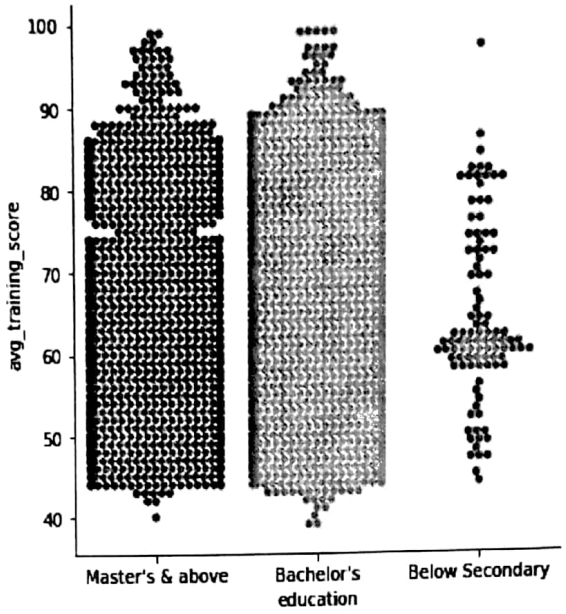
Sworm Plot

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

Out[16]: <seaborn.axisgrid.FacetGrid at 0x1c9fae66ba8>



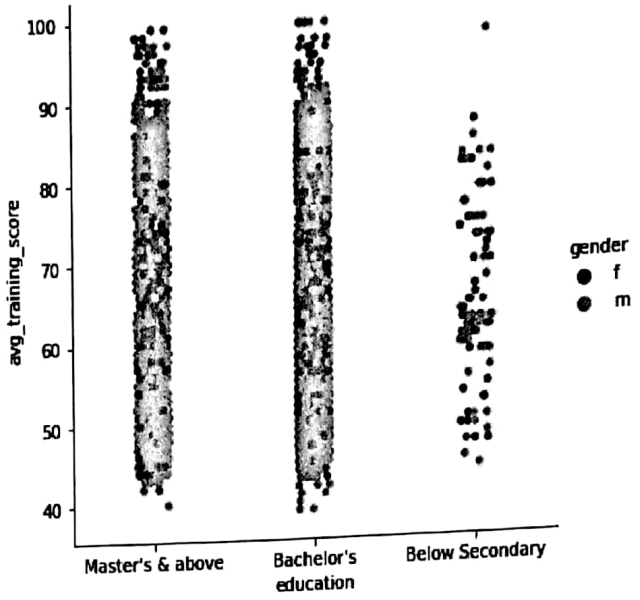
Hue Plot

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

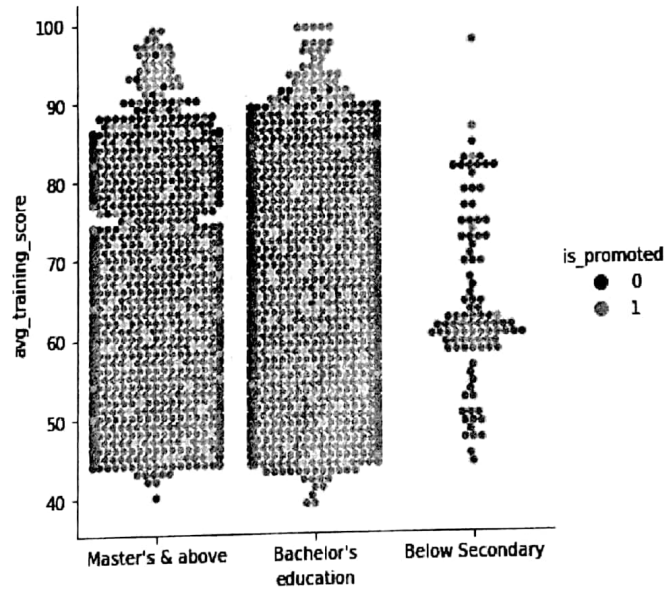
Out[17]: <seaborn.axisgrid.FacetGrid at 0x1c9faed7e80>



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

In [18]: sns.catplot(x="education", y="avg_training_score", hue="is_promoted", kind="swarm",

Out[18]: <seaborn.axisgrid.FacetGrid at 0x1c9faebef28>



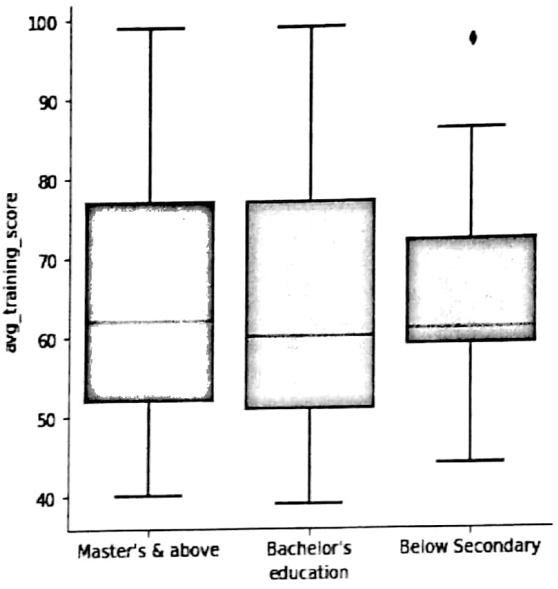
From this plot, we can clearly see people with higher scores got a promotion.

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 [20]: sns.catplot(x="education", y="avg_training_score", kind="box",
                     data=df2)
Out[20]: <seaborn.axisgrid.FacetGrid at 0x1c9f998c208>
```



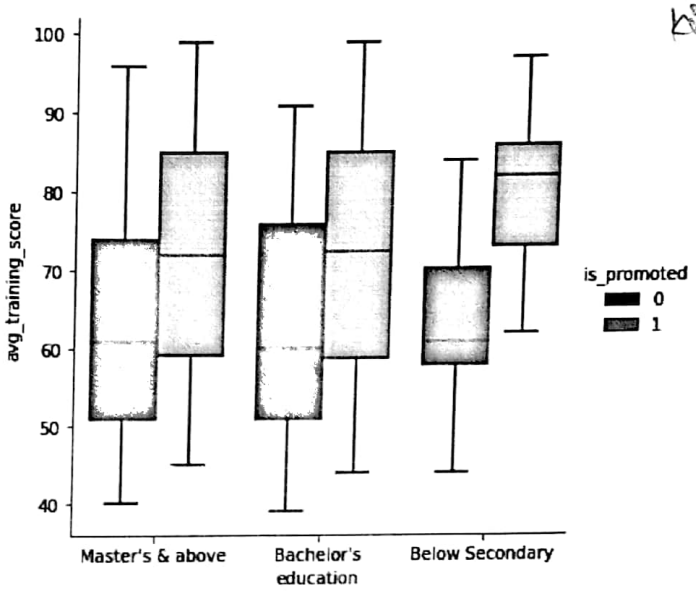
From this chart, we can understand that promotees with masters degree have a minimum of 40, maximum of 100 scores and average score of around 62. Similarly, we can see the 25th and 75th percetile scores are around 52 and 78. Similarly, we can interpret for bachelors and below secondary categories as well.

Box Plot with Hue Dimension

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

In [21]:
```
sns.catplot(x="education", y="avg_training_score",
hue="is_promoted",
kind="box")
```

Out[21]: <seaborn.axisgrid.FacetGrid at 0x1c9f9837e80>



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.

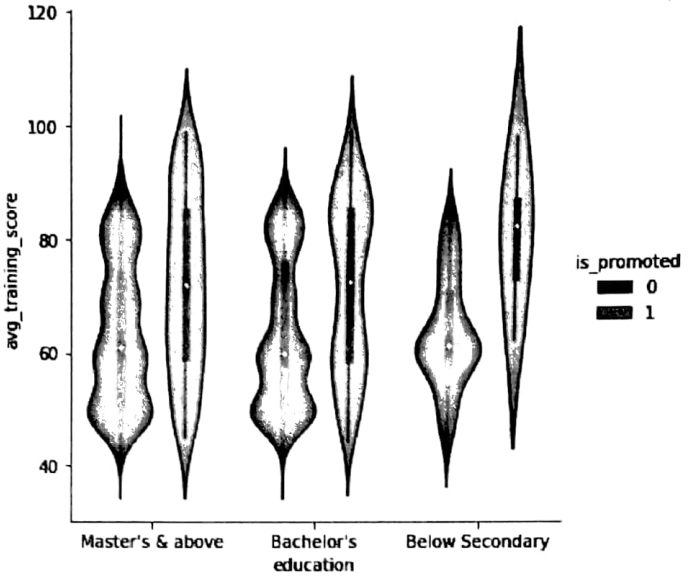
Violin Plot

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 [22]: sns.catplot(x="education", y="avg_training_score", hue="is_promoted", kind="violin")

Out[22]: <seaborn.axisgrid.FacetGrid at 0x1c9f9bb5160>

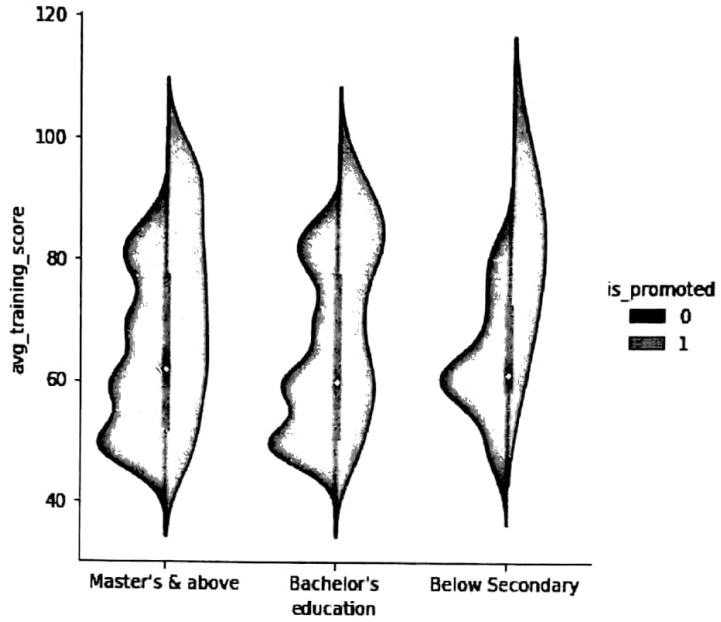


We can see in the above violin plot that each education category is represented with 2 violins one for promoted and the other not promoted target. We can also split the violin when the hue semantic parameter has only two levels, which could also be helpful in saving space on the plot.

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

In [23]: sns.catplot(x="education", y="avg_training_score", hue="is_promoted", kind="violin")

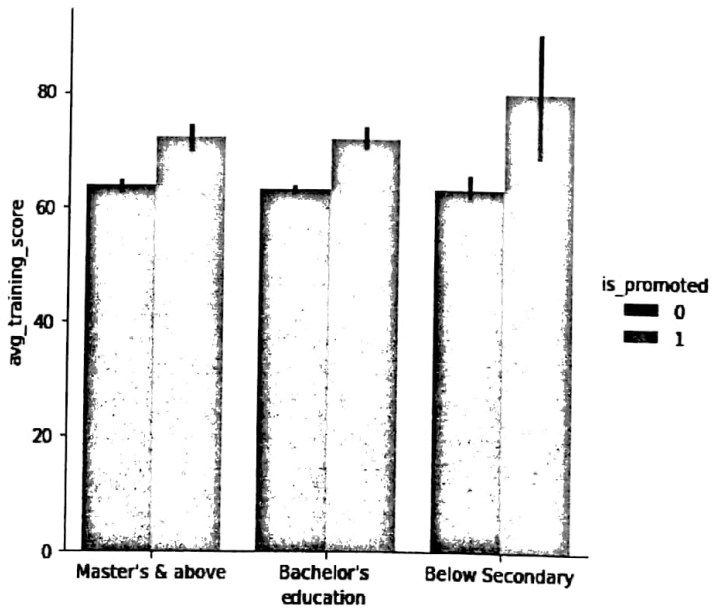
Out[23]: <seaborn.axisgrid.FacetGrid at 0x1c9fced0ef0>



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

In [24]: sns.catplot(x="education", y="avg_training_score", hue="is_promoted", kind="bar")

Out[24]: <seaborn.axisgrid.FacetGrid at 0x1c9fcedf0b8>

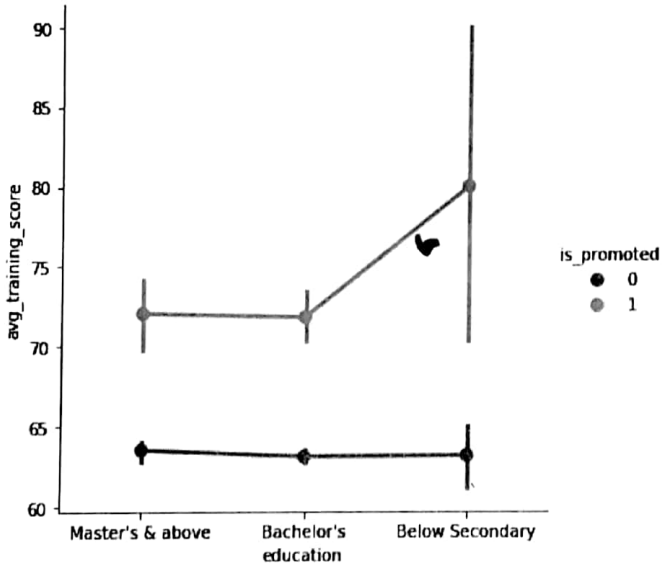


Point Plot

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")
Out[26]: <seaborn.axisgrid.FacetGrid at 0x1c9fd06da90>
```



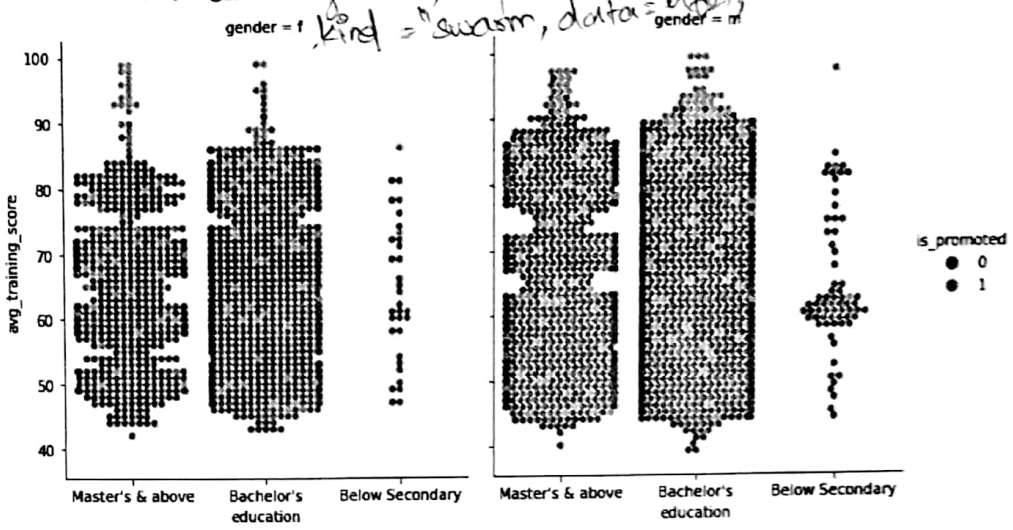
In the above figure, candidates with higher average training score are promoted. Since, we have taken mini dataset with around 700 samples, confidence interval is high for below secondary education level. Graph will show better plot if we take full dataset.

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=df);



3. Visualizing the Distribution of Data

We want to know how data or variables are being distributed. Distribution of data could tell us a lot about the nature of the data. Types of distributions are:

- Univariate distribution (involves one variable)
- Bivariate distribution (involves two variables)

Types of plots

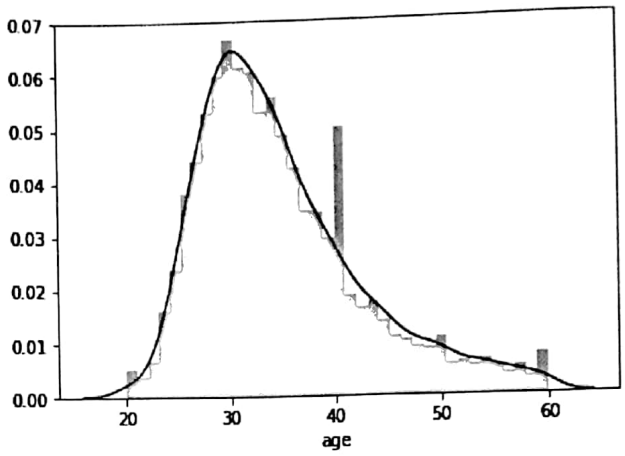
- For univariate distributions: Histogram
- For bivariate distributions: Hexplot, KDE plot, Boxen plot
- Correlation among all columns: heatmap
- Multiple bivariate distributions: pairplot

Plot Univariate Distributions

Plot Histogram with kernel density estimate value for age attribute

In [28]: sns.distplot(df2.age)

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x1c9fd1fbbe0>

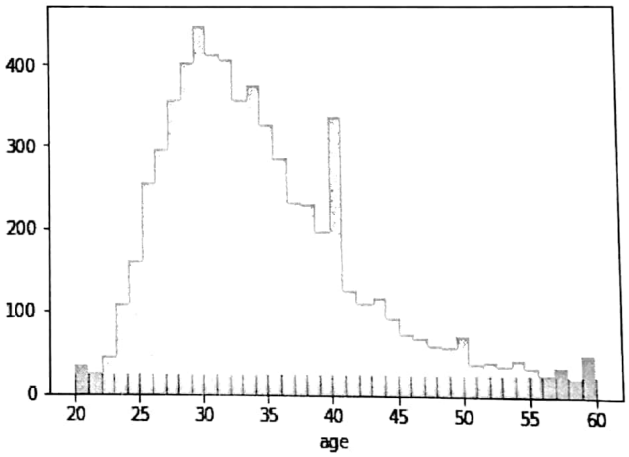


We can understand from .this plot, the average age of candidates. Most of the promotion candidates have age around 25 to 35 years.KDE plot encodes the density of observations (ie., age) on one axis with height along the other axis.

Show only Histogram for age variable, without KDE

In [29]: sns.distplot(df2.age, kde=False, rug=True)

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x1c9fd287160>



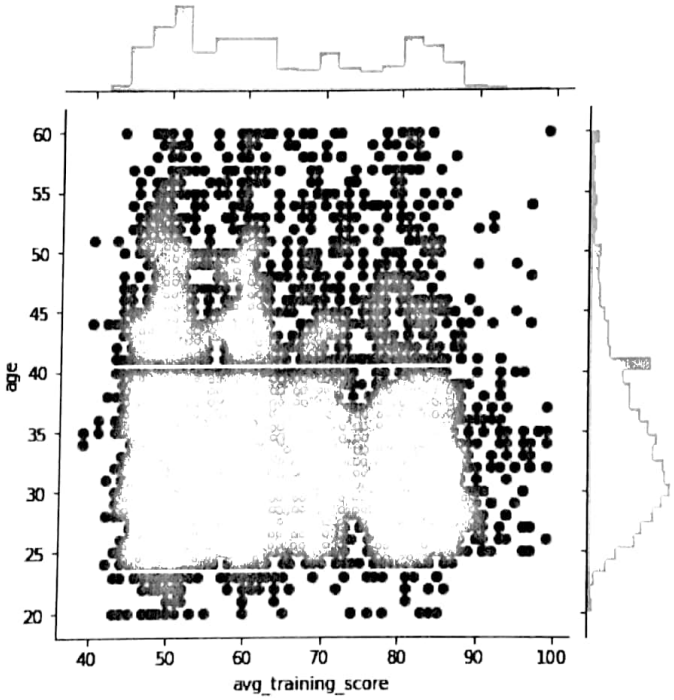
Plot Bivariate Distributions

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=df);
Out[30]: <seaborn.axisgrid.JointGrid at 0x1c9fd2c8630>
```



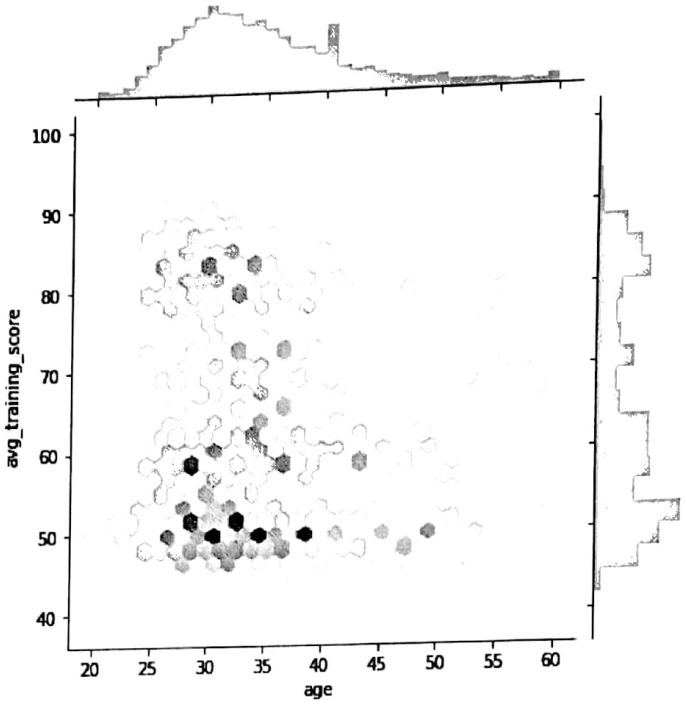
Hex Plot

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 0x1c9fd7bee48>



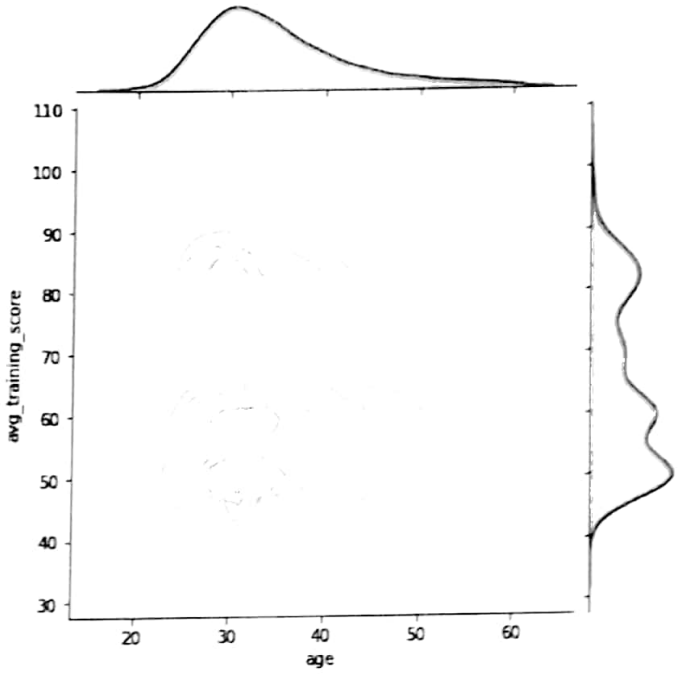
KDE Plot

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 [33]: sns.jointplot(x="age", y="avg_training_score", data=df, kind="kde");

Out[33]: <seaborn.axisgrid.JointGrid at 0x1c9fda503c8>



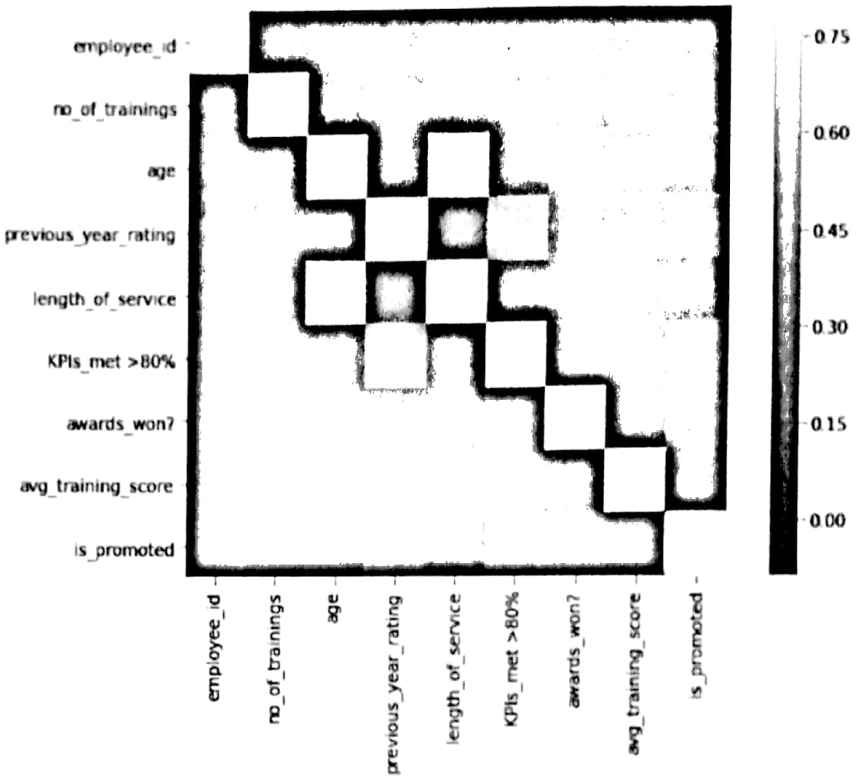
Heat Map

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 [34]:
```
corrmat = df2.corr()
f, ax = plt.subplots(figsize=(9,6))
sns.heatmap(corrmat, vmax=.8, square=True)
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1c9fdb61be0>



Can you answer these questions about the previous heatmap?

- What's the strongest and what's the weakest correlated pair (except the main diagonal)?

- What are the three variables most correlated with the target variable, is_promoted ?

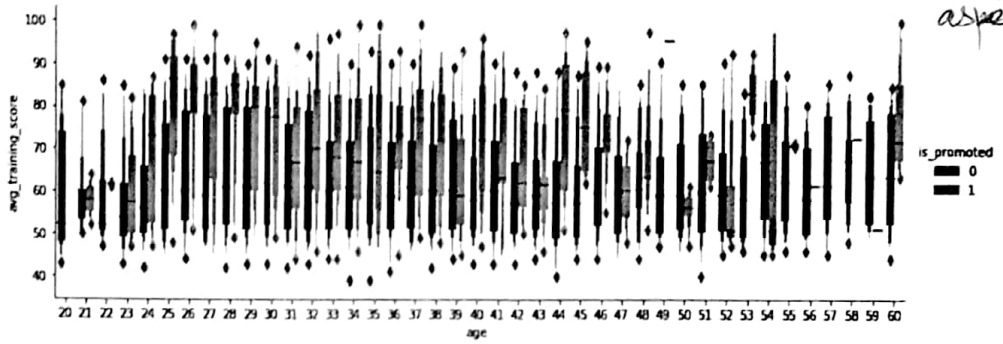
Boxen Plot

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 [35]: sns.catplot(x="age", y="avg_training_score", data=df2,
                     kind="boxen", height=4, aspect=2.7)
Out[35]: <seaborn.axisgrid.FacetGrid at 0x1c9fdc0e9b0>
```



Pair Plot

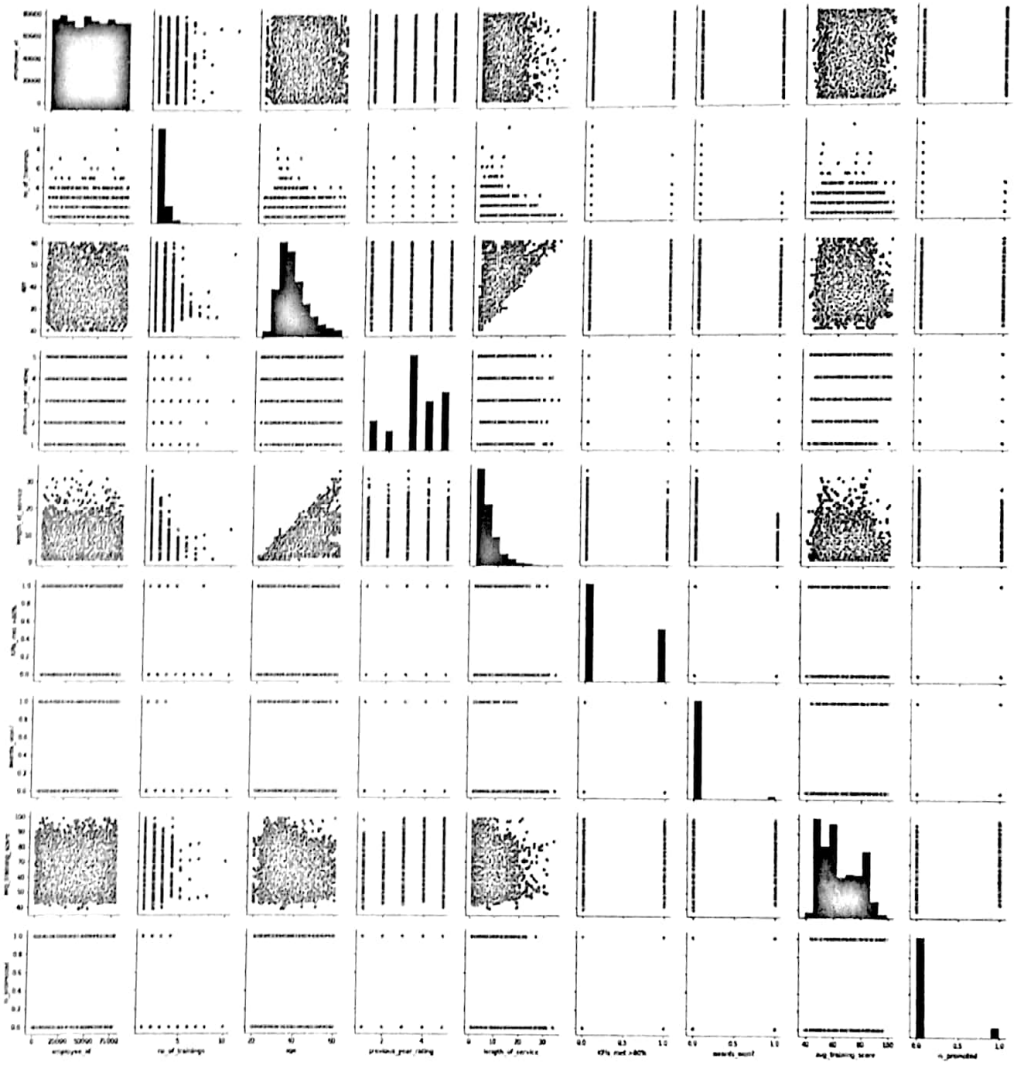
We can also plot multiple bivariate distributions in a dataset by using pairplot() function of the seaborn library. This shows the relationship between each column of the database. It also draws the univariate distribution plot of each variable on the diagonal axis

Draw a Pair Plot for the dataset

In [36]: sns.pairplot(df2)

```
C:\Users\Rajkumar\Anaconda3\lib\site-packages\numpy\lib\histograms.py:839: Ru
ntimeWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\Rajkumar\Anaconda3\lib\site-packages\numpy\lib\histograms.py:840: Ru
ntimeWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

Out[36]: <seaborn.axisgrid.PairGrid at 0x1c9fdb61dd8>



In [ ]: