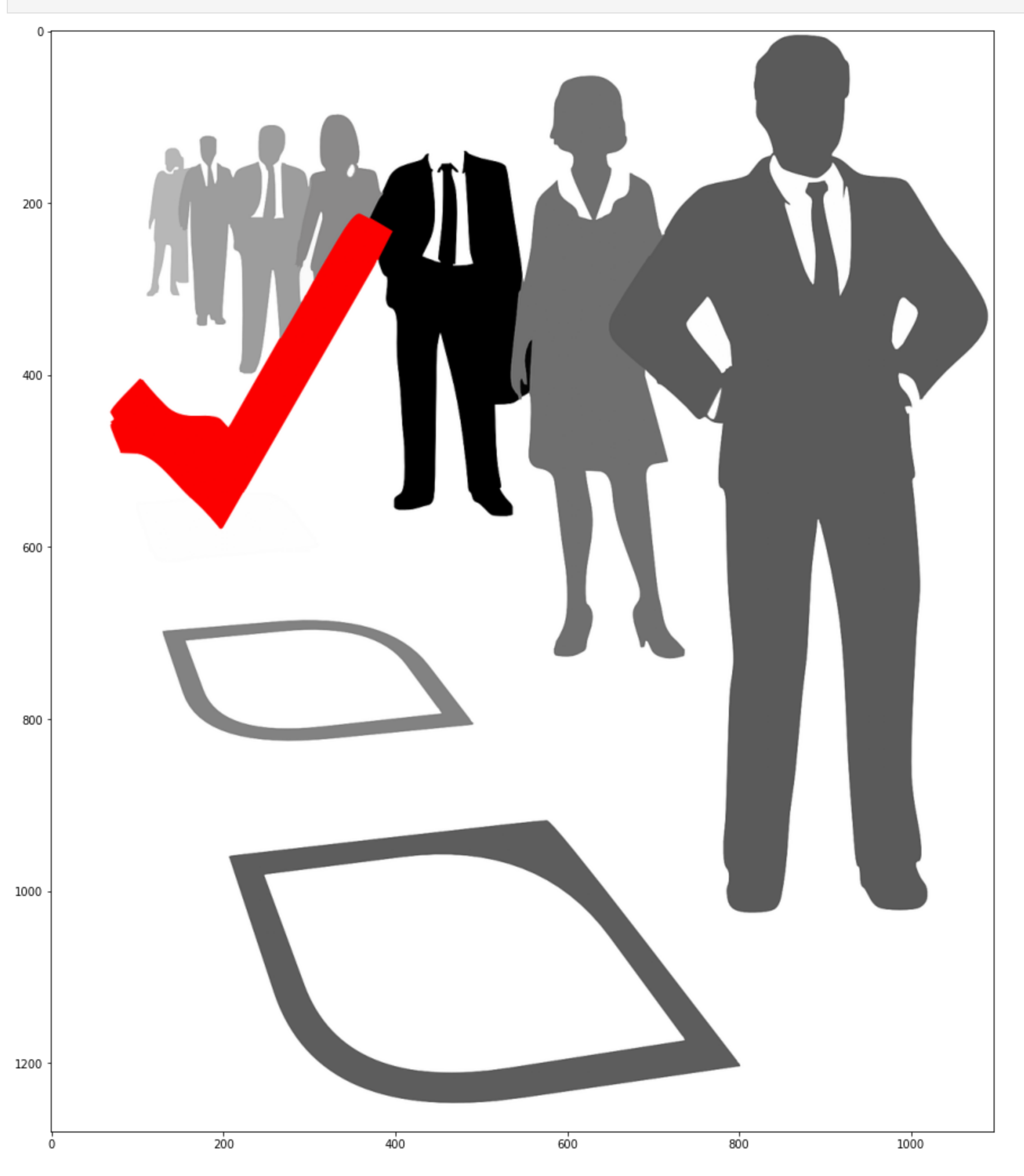
Employee Promotion Model

- Perform a full exploratory data analysis on the datset.
- Find whatever insights you can.
- EDA of the dataset depicting the different analysis which one found and perform operations on the data set like cleaning and encoding.
- Also check is data is alright for data preprocessing.
- Also create visualizations using univariate bivariate and multivariate analysis.
- Create a classification problem model using:

> logistic
>
> decision trees
>
> random forest

In [60]:
```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
plt.figure(figsize=(15,20))
img = mpimg.imread(r'C:\Users\PC-chetan\Downloads\headhunt-gcc00ffffa_1280.png')
imgplot = plt.imshow(img)
```



All Insights before cleaning

- here two data sets are given
- train data and the test data
- in test data column is_ promoted is missing which we have to find
- and is_proomoted is the column of which values we have to pridict
- data is looking good no. of clumns are also less
- data have different type of datas
- #train data have 13 columns
- test data have 12 columns
- train data have 54808 columns and 13 rows
- test data have 23490 columns and 12 rows
- both the data have similar data types
- 7 int dtypes columns
- float dtypes columns
- object dtypes columns
- st off the column have zero null values
- only two column have null vaalues
- no_of_trainings max= 10, min =1 average = 1.25
- age max = 60, min = 20, average = 35 approx.
- previous_year_rating max = 5, min = 1 average = 3.3 approx.
- length_of_service max = 37, min = 1, average = 5.8 approx.
- avg_training_score max = 99, min = 39, average = 63 approx.
- for both train and test data there is no relation b/w the feature of the data

STEP 1

- import the required libraries

In [61]:
```
import pandas as pd# we import pandas to handle the file and its save our lots of time, it handle large data set efficeintly.
import numpy as np #for creating arrays
import seaborn as sns #for data visualzation
import matplotlib.pyplot as plt # for data visualization
```

step 2

-read the given data set

In [62]:
```
df1 = pd.read_csv(r'C:\Users\PC-chetan\Desktop\train.csv') # trian data
```

In [63]:
```
df2 = pd.read_csv(r'C:\Users\PC-chetan\Desktop\test.csv') # test data
```

In [64]:
```
# here two data sets are given
# train data and the test data
# in test data column is_ promoted is missing because the data is alredy split in the two parts
# and is_proomoted is the column of which values we have to pridict
```

step 3

- now we'll do general analysis of the data.

Descriptive Statistics

- It gives us Summary about all the continuous and Categorical Variables present in the dataset.

Exploration of the data

- for understainding the nature of the data and finding out the following:-
- no. of columns. no. of rows. shape of the data. index of the data. data type of the each feature(column).
- find the list of not use full colmns. no. of null values in each column. after that we will see some vizuals of our data before data cleaning.

In [65]:
```
df1.head()
```

Out[65]:

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	awards_won?	avg_training_score	is_promoted
0	65438	Sales & Marketing	region_7	Master's & above	f	sourcing	1	35	5.0	8	0	49	0
1	65141	Operations	region_22	Bachelor's	m	other	1	30	5.0	4	0	60	0
2	7513	Sales & Marketing	region_19	Bachelor's	m	sourcing	1	34	3.0	7	0	50	0
3	2542	Sales & Marketing	region_23	Bachelor's	m	other	2	39	1.0	10	0	50	0
4	48945	Technology	region_26	Bachelor's	m	other	1	45	3.0	2	0	73	0

In [66]:
```
df2.head()
```

Out[66]:

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	awards_won?	avg_training_score
0	8724	Technology	region_26	Bachelor's	m	sourcing	1	24	NaN	1	0	77
1	74430	HR	region_4	Bachelor's	f	other	1	31	3.0	5	0	51
2	72255	Sales & Marketing	region_13	Bachelor's	m	other	1	31	1.0	4	0	47
3	38562	Procurement	region_2	Bachelor's	f	other	3	31	2.0	9	0	65
4	64486	Finance	region_29	Bachelor's	m	sourcing	1	30	4.0	7	0	61

In [67]:
```
df1.tail()
```

Out[67]:

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	awards_won?	avg_training_score	is_promoted
54803	3030	Technology	region_14	Bachelor's	m	sourcing	1	48	3.0	17	0	78	0
54804	74592	Operations	region_27	Master's & above	f	other	1	37	2.0	6	0	56	0
54805	13918	Analytics	region_1	Bachelor's	m	other	1	27	5.0	3	0	79	0
54806	13614	Sales & Marketing	region_9	NaN	m	sourcing	1	29	1.0	2	0	45	0
54807	51526	HR	region_22	Bachelor's	m	other	1	27	1.0	5	0	49	0

In [68]:
```
df2.tail()
```

Out[68]:

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	awards_won?	avg_training_score
23485	53478	Legal	region_2	Below Secondary	m	sourcing	1	24	3.0	1	0	61
23486	25600	Technology	region_25	Bachelor's	m	sourcing	1	31	3.0	7	0	74
23487	45409	HR	region_16	Bachelor's	f	sourcing	1	26	4.0	4	0	50
23488	1186	Procurement	region_31	Bachelor's	m	sourcing	3	27	NaN	1	0	70
23489	5973	Technology	region_17	Master's & above	m	other	3	40	5.0	5	0	89

In [69]:
```
# data is looking good no. of clumns are also less
# data have different type of datas
```

In [70]:
```
df1.columns
```

Out[70]:
```
Index(['employee_id', 'department', 'region', 'education', 'gender',
       'recruitment_channel', 'no_of_trainings', 'age', 'previous_year_rating',
       'length_of_service', 'awards_won?', 'avg_training_score',
       'is_promoted'],
      dtype='object')
```

In [71]:
```
#train data have 13 columns
```

In [72]:
```
df2.columns
```

Out[72]:
```
Index(['employee_id', 'department', 'region', 'education', 'gender',
       'recruitment_channel', 'no_of_trainings', 'age', 'previous_year_rating',
       'length_of_service', 'awards_won?', 'avg_training_score'],
      dtype='object')
```

In [73]:
```
# test data have 12 columns
```

In [74]:
```
df2.columns.value_counts()
```

Out[74]:
```
employee_id             1
department              1
region                  1
education               1
gender                  1
recruitment_channel     1
no_of_trainings         1
age                     1
previous_year_rating    1
length_of_service       1
awards_won?             1
avg_training_score      1
dtype: int64
```

In [75]:
```
df1.shape
```

Out[75]:
```
(54808, 13)
```

In [76]:
```
#train data have 54808 columns and 13 rows
```

In [77]:
```
df2.shape
```

Out[77]:
```
(23490, 12)
```

In [78]:
```
#test data have 23490 columns and 12 rows
```

In [79]:
```
df1.index
```

Out[79]:
```
RangeIndex(start=0, stop=54808, step=1)
```

In [80]:
```
df2.index
```

Out[80]:
```
RangeIndex(start=0, stop=23490, step=1)
```

In [81]:
```
df1.dtypes
```

Out[81]:
```
employee_id               int64
department               object
region                   object
education                object
gender                   object
recruitment_channel      object
no_of_trainings           int64
age                       int64
previous_year_rating    float64
length_of_service         int64
awards_won?               int64
avg_training_score        int64
is_promoted               int64
dtype: object
```

In [82]:
```
# both the data have similar data types
# 7 int  dtypes columns
# 1 float dtypes columns
# 5 object dtypes columns
```

In [83]:
```
df2.dtypes
```

Out[83]:
```
employee_id               int64
department               object
region                   object
education                object
gender                   object
recruitment_channel      object
no_of_trainings           int64
age                       int64
previous_year_rating    float64
length_of_service         int64
awards_won?               int64
avg_training_score        int64
dtype: object
```

In [84]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54808 entries, 0 to 54807
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   employee_id           54808 non-null  int64
 1   department            54808 non-null  object
 2   region                54808 non-null  object
 3   education             52399 non-null  object
 4   gender                54808 non-null  object
 5   recruitment_channel   54808 non-null  object
 6   no_of_trainings       54808 non-null  int64
 7   age                   54808 non-null  int64
 8   previous_year_rating  50684 non-null  float64
 9   length_of_service     54808 non-null  int64
 10  awards_won?           54808 non-null  int64
 11  avg_training_score    54808 non-null  int64
 12  is_promoted           54808 non-null  int64
dtypes: float64(1), int64(7), object(5)
memory usage: 5.4+ MB
```

In [85]:
```
# most off the column have zero null values
# only two column have null vaalues
```

In [86]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23490 entries, 0 to 23489
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   employee_id           23490 non-null  int64
 1   department            23490 non-null  object
 2   region                23490 non-null  object
 3   education             22456 non-null  object
 4   gender                23490 non-null  object
 5   recruitment_channel   23490 non-null  object
 6   no_of_trainings       23490 non-null  int64
 7   age                   23490 non-null  int64
 8   previous_year_rating  21678 non-null  float64
 9   length_of_service     23490 non-null  int64
 10  awards_won?           23490 non-null  int64
 11  avg_training_score    23490 non-null  int64
dtypes: float64(1), int64(6), object(5)
memory usage: 2.2+ MB
```

In [87]:
```
df1.describe().style.background_gradient(axis=0)
```

Out[87]:

	employee_id	no_of_trainings	age	previous_year_rating	length_of_service	awards_won?	avg_training_score	is_promoted
count	54808.000000	54808.000000	54808.000000	50684.000000	54808.000000	54808.000000	54808.000000	54808.000000
mean	39195.830627	1.253011	34.803915	3.329256	5.865512	0.023172	63.386750	0.085170
std	22586.581449	0.609264	7.660169	1.259993	4.265094	0.150450	13.371559	0.279137
min	1.000000	1.000000	20.000000	1.000000	1.000000	0.000000	39.000000	0.000000
25%	19669.750000	1.000000	29.000000	3.000000	3.000000	0.000000	51.000000	0.000000
50%	39225.500000	1.000000	33.000000	3.000000	5.000000	0.000000	60.000000	0.000000
75%	58730.500000	1.000000	39.000000	4.000000	7.000000	0.000000	76.000000	0.000000
max	78298.000000	10.000000	60.000000	5.000000	37.000000	1.000000	99.000000	1.000000

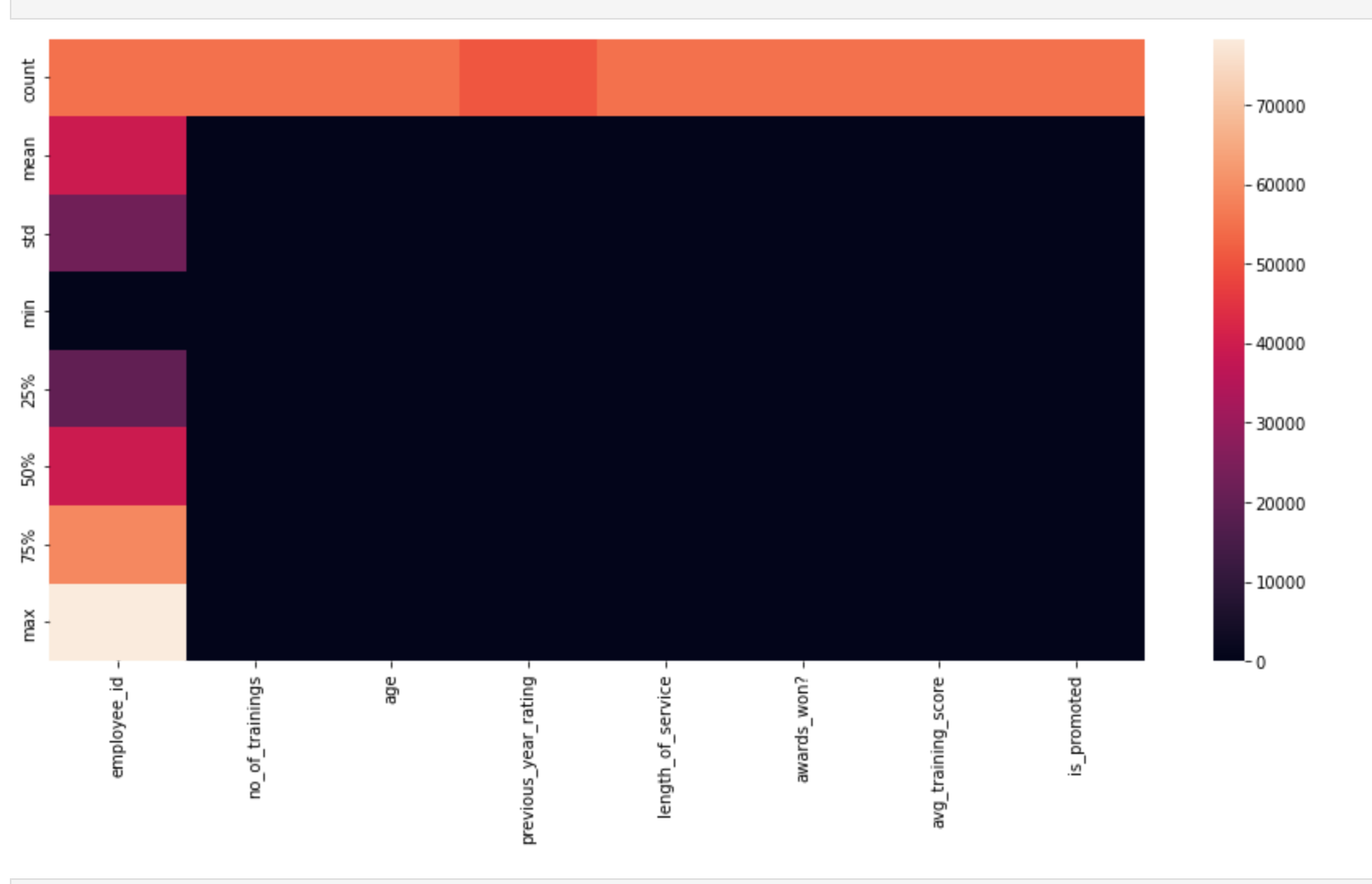
In [88]:
```
# no_of_trainings max= 10, min =1  average = 1.25
# age max = 60, min = 20, average = 35 approx.
# previous_year_rating max = 5, min = 1 average = 3.3 approx.
# length_of_service max = 37, min = 1, average = 5.8 approx.
# avg_training_score max = 99, min = 39, average = 63 approx.
```

In [89]:
```
df2.describe().T.style.background_gradient(axis=1)
```

Out[89]:

	count	mean	std	min	25%	50%	75%	max
employee_id	23490.000000	39041.399149	22640.809201	3.000000	19370.250000	38963.500000	58690.000000	78295.000000
no_of_trainings	23490.000000	1.254236	0.600910	1.000000	1.000000	1.000000	1.000000	9.000000
age	23490.000000	34.782929	7.679492	20.000000	29.000000	33.000000	39.000000	60.000000
previous_year_rating	21678.000000	3.339146	1.263294	1.000000	3.000000	3.000000	4.000000	5.000000
length_of_service	23490.000000	5.810387	4.207917	1.000000	3.000000	5.000000	7.000000	34.000000
awards_won?	23490.000000	0.022776	0.149191	0.000000	0.000000	0.000000	0.000000	1.000000
avg_training_score	23490.000000	63.263133	13.411750	39.000000	51.000000	60.000000	76.000000	99.000000

In [90]:
```
plt.figure(figsize= (15,7))
sns.heatmap(df1.describe())
plt.show()
```



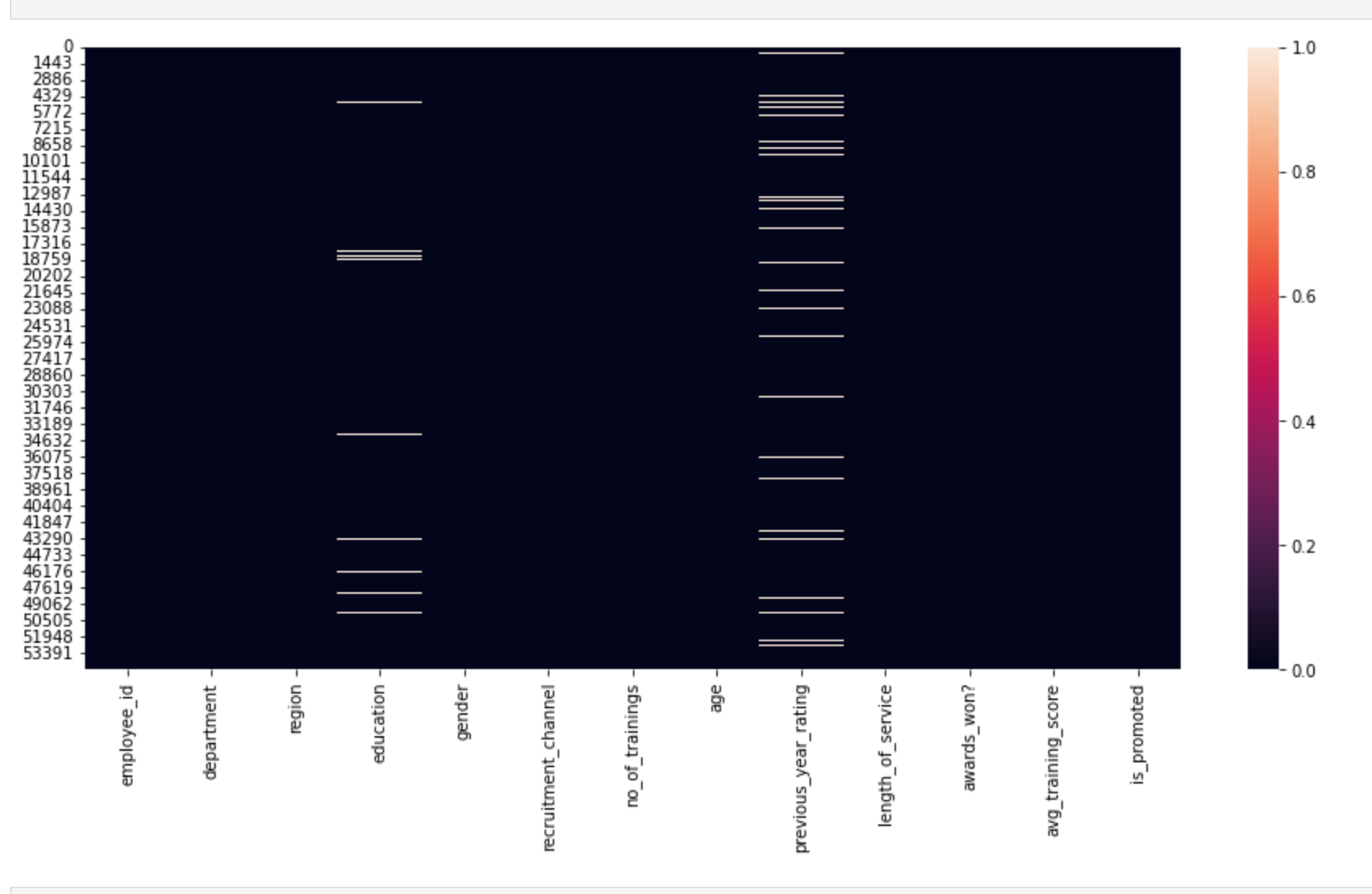
In [91]:
```
df1.isnull().sum()
```

Out[91]:
```
employee_id                0
department                 0
region                     0
education               2409
gender                     0
recruitment_channel        0
no_of_trainings            0
age                        0
previous_year_rating    4124
length_of_service          0
awards_won?                0
avg_training_score         0
is_promoted                0
dtype: int64
```

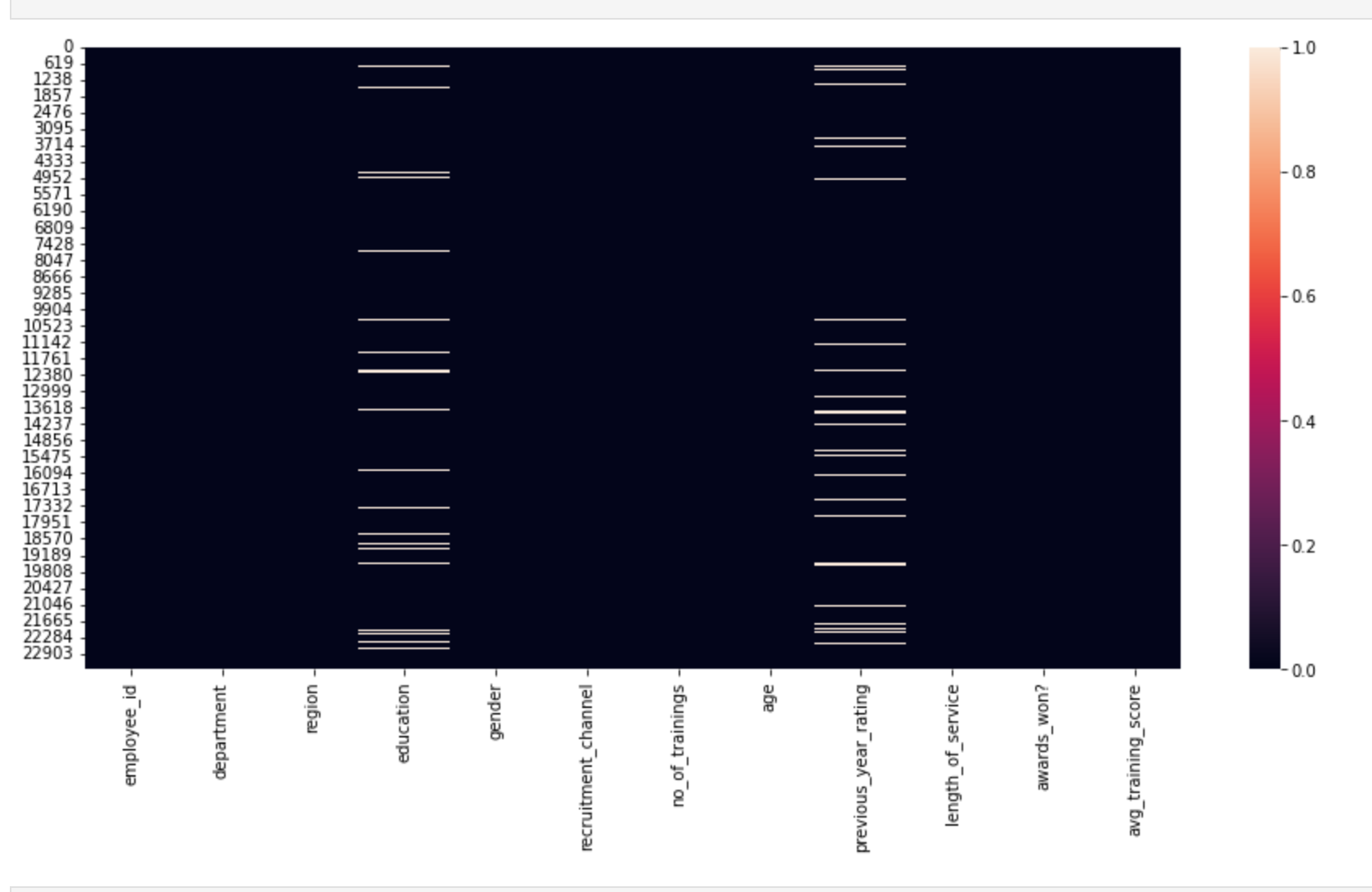
In [92]:
```
df2.isnull().sum()
```

Out[92]:
```
employee_id                0
department                 0
region                     0
education               1034
gender                     0
recruitment_channel        0
no_of_trainings            0
age                        0
previous_year_rating    1812
length_of_service          0
awards_won?                0
avg_training_score         0
dtype: int64
```

In [93]:
```
plt.figure(figsize= (15,7))
sns.heatmap(df1.isnull())
plt.show()
```



In [94]:
```
plt.figure(figsize= (15,7))
sns.heatmap(df2.isnull())
plt.show()
```

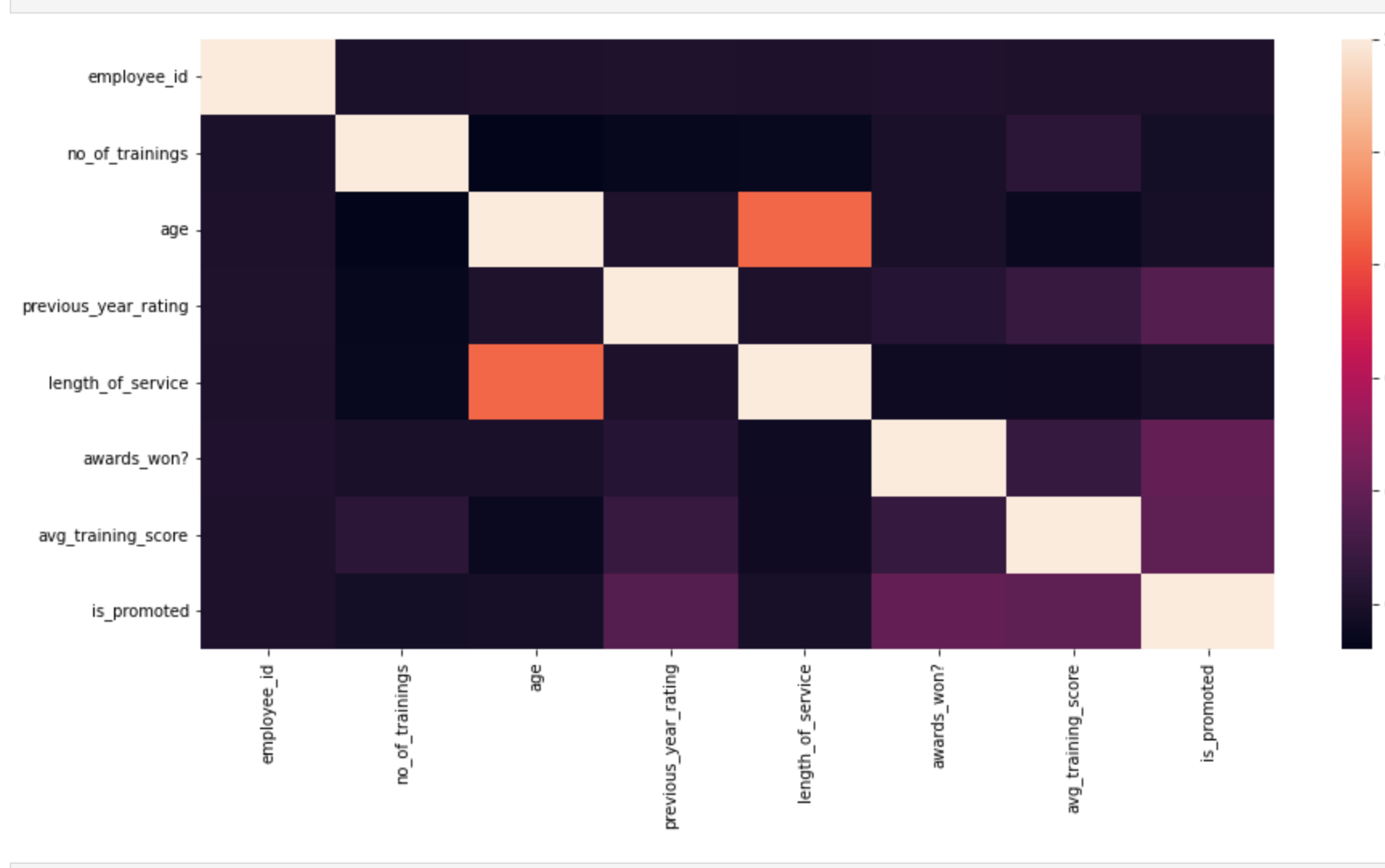


In [95]:
```
df1.corr()
```

Out[95]:

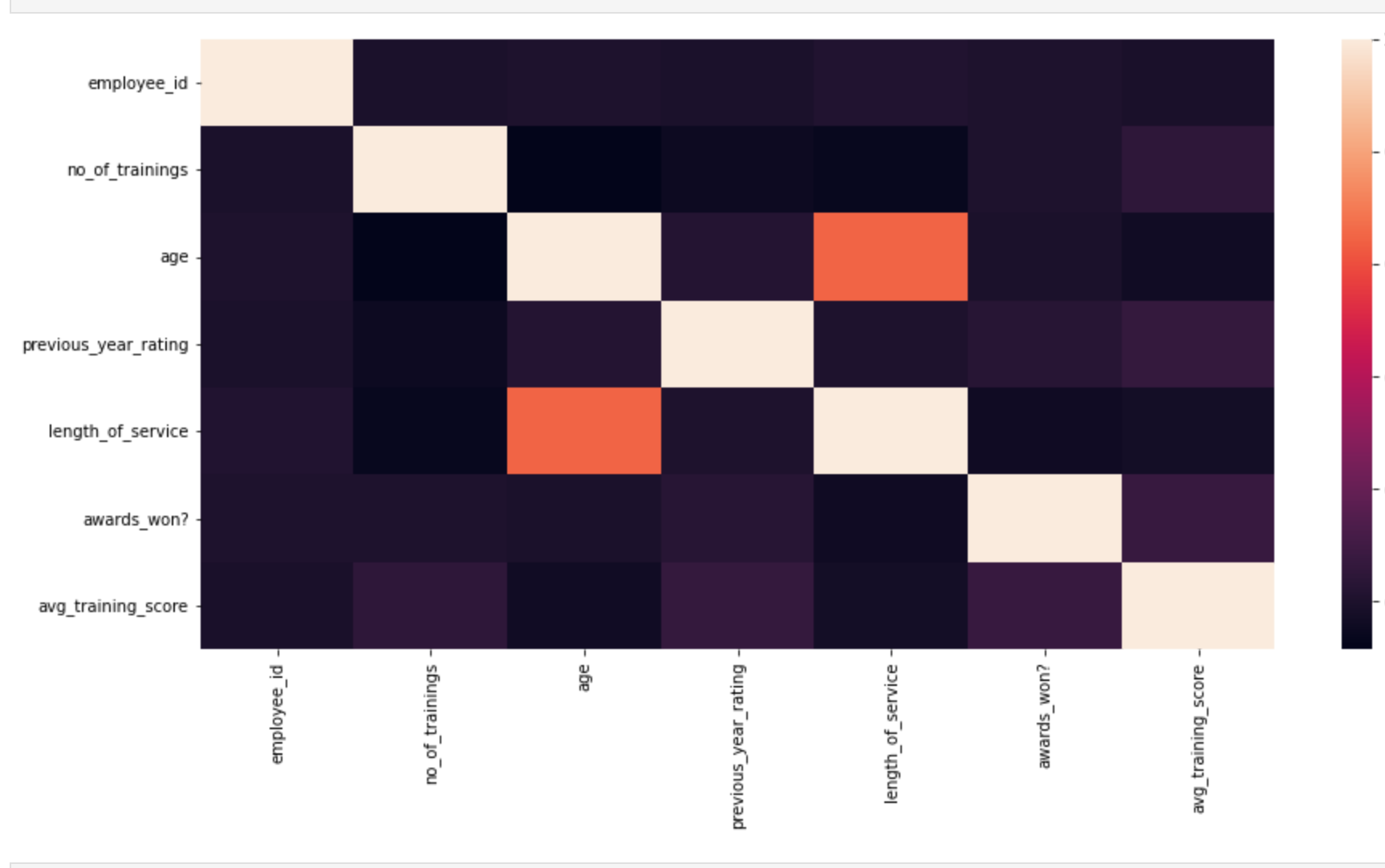
	employee_id	no_of_trainings	age	previous_year_rating	length_of_service	awards_won?	avg_training_score	is_promoted
employee_id	1.000000	-0.005121	0.000437	0.004533	0.001274	0.008420	-0.000586	0.001206
no_of_trainings	-0.005121	1.000000	-0.081278	-0.063126	-0.057275	-0.007628	0.042517	-0.024896
age	0.000437	-0.081278	1.000000	0.006008	0.657111	-0.008169	-0.048380	-0.017166
previous_year_rating	0.004533	-0.063126	0.006008	1.000000	0.000253	0.027738	0.075139	0.159320
length_of_service	0.001274	-0.057275	0.657111	0.000253	1.000000	-0.039927	-0.038122	-0.010670
awards_won?	0.008420	-0.007628	-0.008169	0.027738	-0.039927	1.000000	0.072138	0.195871
avg_training_score	-0.000586	0.042517	-0.048380	0.075139	-0.038122	0.072138	1.000000	0.181147
is_promoted	0.001206	-0.024896	-0.017166	0.159320	-0.010670	0.195871	0.181147	1.000000

In [96]:
```
plt.figure(figsize= (15,7))
sns.heatmap(df1.corr())
plt.show()
```



In [97]:
```
#for both train and test data there is no relation b/w the feature of the data
```

In [98]:
```
plt.figure(figsize= (15,7))
sns.heatmap(df2.corr())
plt.show()
```



In [ ]: