In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,StandardScaler
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [4]:

```
df = pd.read_csv("test.csv")
```

In [5]:

```
df
```

Out[5]:

| | employee_id | department | region | education | gender | recruitment_channel | no_of_t |
|---|---|---|---|---|---|---|---|
| **0** | 8724 | Technology | region_26 | Bachelor's | m | sourcing | |
| **1** | 74430 | HR | region_4 | Bachelor's | f | other | |
| **2** | 72255 | Sales & Marketing | region_13 | Bachelor's | m | other | |
| **3** | 38562 | Procurement | region_2 | Bachelor's | f | other | |
| **4** | 64486 | Finance | region_29 | Bachelor's | m | sourcing | |
| **...** | ... | ... | ... | ... | ... | ... | |
| **23485** | 53478 | Legal | region_2 | Below Secondary | m | sourcing | |
| **23486** | 25600 | Technology | region_25 | Bachelor's | m | sourcing | |
| **23487** | 45409 | HR | region_16 | Bachelor's | f | sourcing | |
| **23488** | 1186 | Procurement | region_31 | Bachelor's | m | sourcing | |
| **23489** | 5973 | Technology | region_17 | Master's & above | m | other | |

23490 rows × 12 columns

In [9]:

```
df.shape
```

Out[9]:

```
(23490, 12)
```

In [10]:

```
df.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 [11]:

```
df.columns
```

Out[11]:

```
Index(['employee_id', 'department', 'region', 'education', 'gender',
       'recruitment_channel', 'no_of_trainings', 'age', 'previous_year_rat
ing',
       'length_of_service', 'awards_won?', 'avg_training_score'],
      dtype='object')
```

In [12]:

```
df.head()
```

Out[12]:

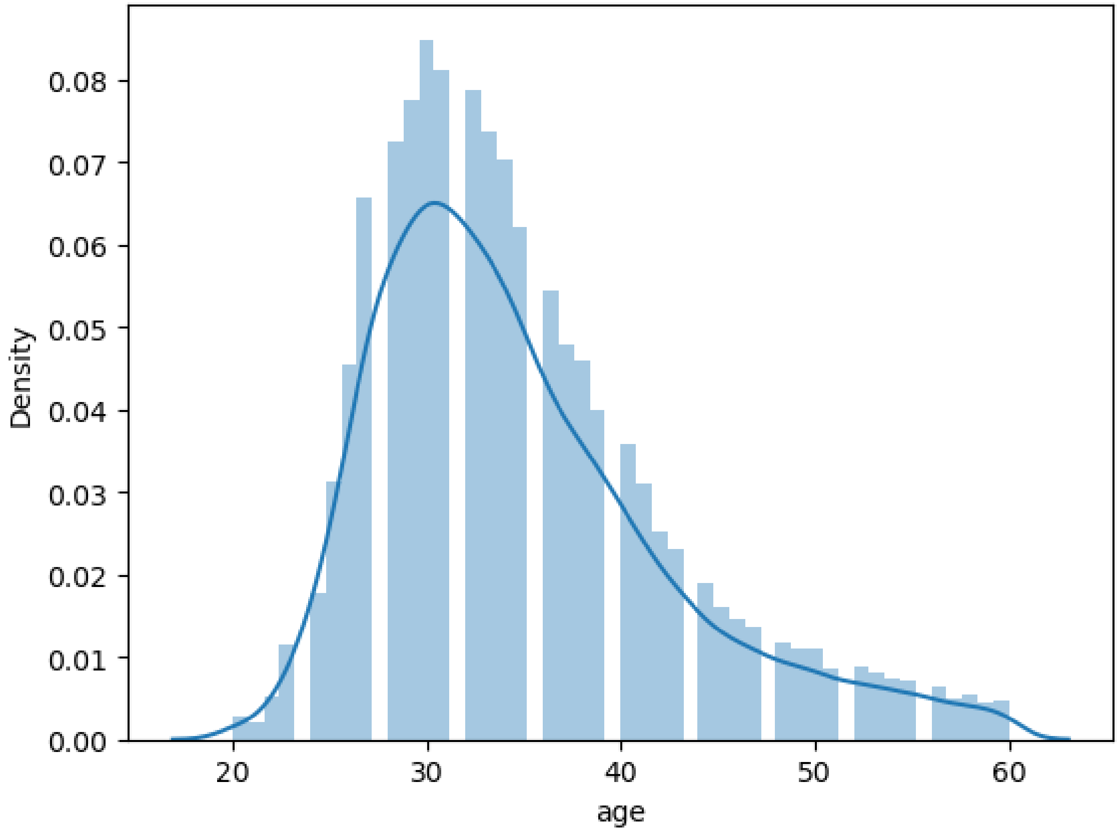
| | employee_id | department | region | education | gender | recruitment_channel | no_of_traini |
|---|---|---|---|---|---|---|---|
| **0** | 8724 | Technology | region_26 | Bachelor's | m | sourcing | |
| **1** | 74430 | HR | region_4 | Bachelor's | f | other | |
| **2** | 72255 | Sales & Marketing | region_13 | Bachelor's | m | other | |
| **3** | 38562 | Procurement | region_2 | Bachelor's | f | other | |
| **4** | 64486 | Finance | region_29 | Bachelor's | m | sourcing | |

In [17]:

```
sns.distplot(df["age"])
plt.show
```

Out[17]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



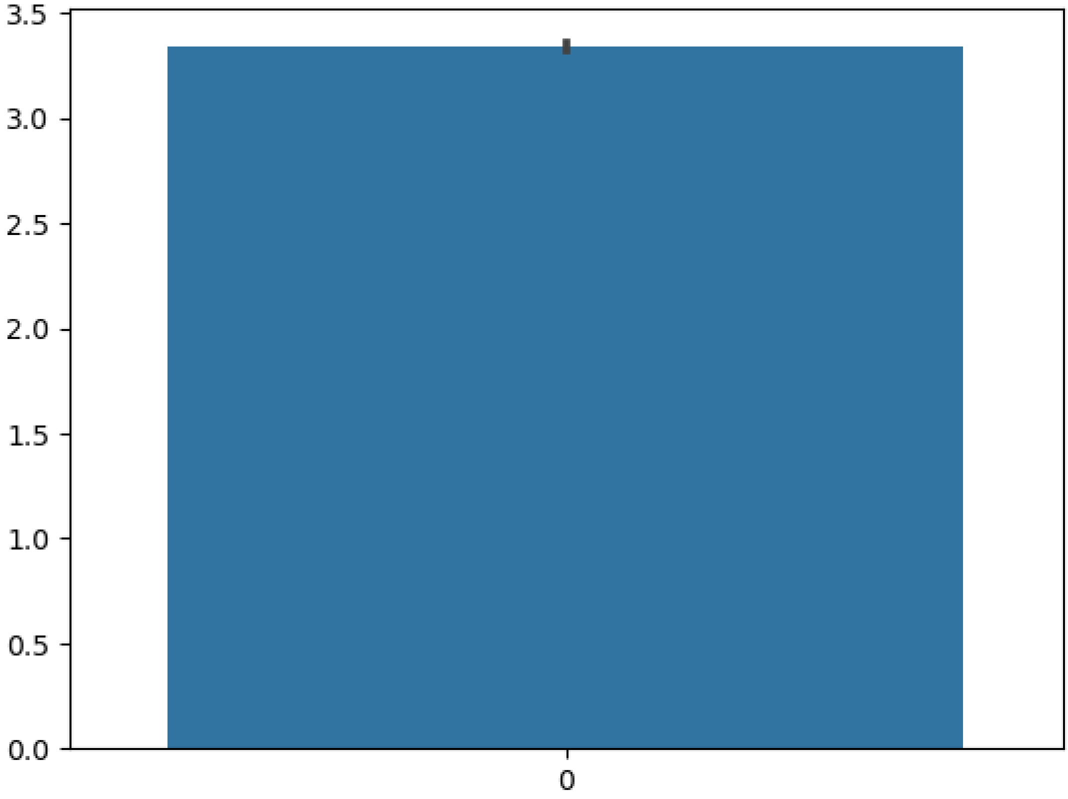
In [18]:

```
dept = df.iloc[:,[5,27]].copy()
dept_per = dept.copy()
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Cell In[18], line 1
----> 1 dept = df.iloc[:,[5,27]].copy()
      2 dept_per = dept.copy()

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1067, in _Locat
ionIndexer.__getitem__(self, key)
   1065     if self._is_scalar_access(key):
   1066         return self.obj._get_value(*key, takeable=self._takeable)
-> 1067     return self._getitem_tuple(key)
   1068 else:
   1069     # we by definition only have the 0th axis
   1070     axis = self.axis or 0

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1563, in _iLocI
ndexer._getitem_tuple(self, tup)
   1561 def _getitem_tuple(self, tup: tuple):
-> 1563     tup = self._validate_tuple_indexer(tup)
   1564     with suppress(IndexingError):
   1565         return self._getitem_lowerdim(tup)

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:873, in _Locati
onIndexer._validate_tuple_indexer(self, key)
    871 for i, k in enumerate(key):
    872     try:
--> 873         self._validate_key(k, i)
    874     except ValueError as err:
    875         raise ValueError(
    876             "Location based indexing can only have "
    877             f"[{self._valid_types}] types"
    878         ) from err

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1481, in _iLocI
ndexer._validate_key(self, key, axis)
   1479     # check that the key does not exceed the maximum size of the i
ndex
   1480     if len(arr) and (arr.max() >= len_axis or arr.min() < -len_axi
s):
-> 1481         raise IndexError("positional indexers are out-of-bounds")
   1482 else:
   1483     raise ValueError(f"Can only index by location with a [{self._v
alid_types}]")

IndexError: positional indexers are out-of-bounds
```

In [21]:

```
sns.barplot(df['previous_year_rating'])
plt.show()
```



In [23]:

```
enc = LabelEncoder()
for i in (2,3,4,5,6,7,16,26):
    df.iloc[:,i] = enc.fit_transform(df.iloc[:,i])
data.head()
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Cell In[23], line 3
      1 enc = LabelEncoder()
      2 for i in (2,3,4,5,6,7,16,26):
----> 3     df.iloc[:,i] = enc.fit_transform(df.iloc[:,i])
      4 data.head()

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1067, in _Locat
ionIndexer.__getitem__(self, key)
   1065     if self._is_scalar_access(key):
   1066         return self.obj._get_value(*key, takeable=self._takeable)
-> 1067     return self._getitem_tuple(key)
   1068 else:
   1069     # we by definition only have the 0th axis
   1070     axis = self.axis or 0

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1563, in _iLocI
ndexer._getitem_tuple(self, tup)
   1561 def _getitem_tuple(self, tup: tuple):
-> 1563     tup = self._validate_tuple_indexer(tup)
   1564     with suppress(IndexingError):
   1565         return self._getitem_lowerdim(tup)

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:873, in _Locati
onIndexer._validate_tuple_indexer(self, key)
    871 for i, k in enumerate(key):
    872     try:
--> 873         self._validate_key(k, i)
    874     except ValueError as err:
    875         raise ValueError(
    876             "Location based indexing can only have "
    877             f"[{self._valid_types}] types"
    878         ) from err

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1466, in _iLocI
ndexer._validate_key(self, key, axis)
   1464     return
   1465 elif is_integer(key):
-> 1466     self._validate_integer(key, axis)
   1467 elif isinstance(key, tuple):
   1468     # a tuple should already have been caught by this point
   1469     # so don't treat a tuple as a valid indexer
   1470     raise IndexingError("Too many indexers")

File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1557, in _iLocI
ndexer._validate_integer(self, key, axis)
   1555 len_axis = len(self.obj._get_axis(axis))
   1556 if key >= len_axis or key < -len_axis:
-> 1557     raise IndexError("single positional indexer is out-of-bounds")

IndexError: single positional indexer is out-of-bounds
```

In [24]:

```
df.corr()
```

Out[24]:

| | employee_id | region | education | gender | recruitment_channel | no_ |
|---|---|---|---|---|---|---|
| **employee_id** | 1.000000 | -0.005705 | 0.007975 | 0.002769 | 0.000626 | |
| **region** | -0.005705 | 1.000000 | -0.011746 | 0.041971 | 0.000208 | |
| **education** | 0.007975 | -0.011746 | 1.000000 | 0.002660 | -0.008417 | |
| **gender** | 0.002769 | 0.041971 | 0.002660 | 1.000000 | 0.000781 | |
| **recruitment_channel** | 0.000626 | 0.000208 | -0.008417 | 0.000781 | 1.000000 | |
| **no_of_trainings** | -0.005433 | 0.001447 | -0.042698 | 0.084591 | -0.005526 | |
| **age** | 0.000920 | -0.095967 | 0.234119 | -0.021359 | -0.014599 | |
| **previous_year_rating** | -0.005963 | -0.003674 | -0.004603 | -0.022358 | 0.006124 | |
| **length_of_service** | 0.011781 | -0.068306 | 0.151010 | -0.013268 | -0.009340 | |
| **awards_won?** | 0.000118 | 0.013245 | -0.000967 | -0.003750 | -0.000875 | |
| **avg_training_score** | -0.011735 | 0.031382 | -0.009197 | -0.025590 | -0.018367 | |

In [ ]: