IMPORTING LIBRARIES FOR EDA

In [1]:
```
import numpy as np
import pandas as pd
from pandas_profiling import ProfileReport
```

In [2]:
```
aug_train = pd.read_csv(r"C:\Users\Asus\Desktop\HR-Analytics-main\aug_train.csv")
aug_test = pd.read_csv(r"C:\Users\Asus\Desktop\HR-Analytics-main\aug_test.csv")
```

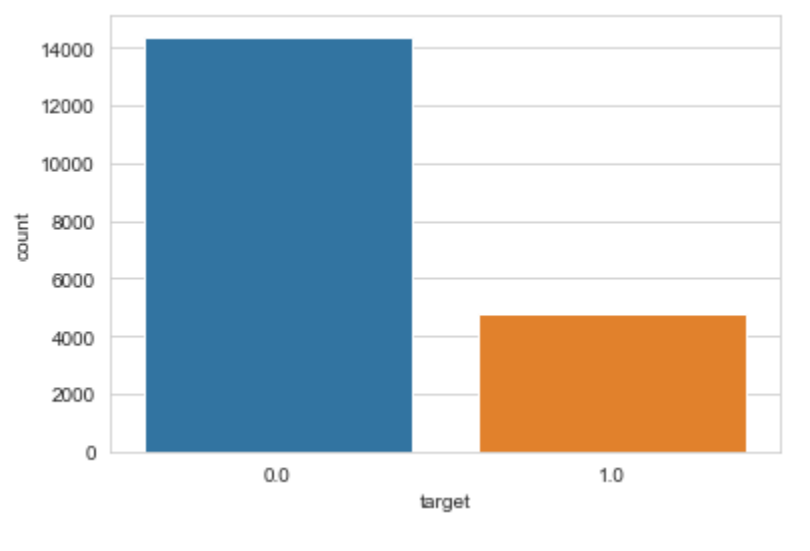
In [3]:
```
profile_train = ProfileReport(aug_train, title="Pandas Profiling Report")
profile_train.to_widgets()
```

In [4]:
```
profile_test = ProfileReport(aug_train, title="Pandas Profiling Report")
profile_test.to_widgets()
```

In [5]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [6]:
```
sns.set_style("whitegrid")
sns.countplot(x='target', data = aug_train)
```

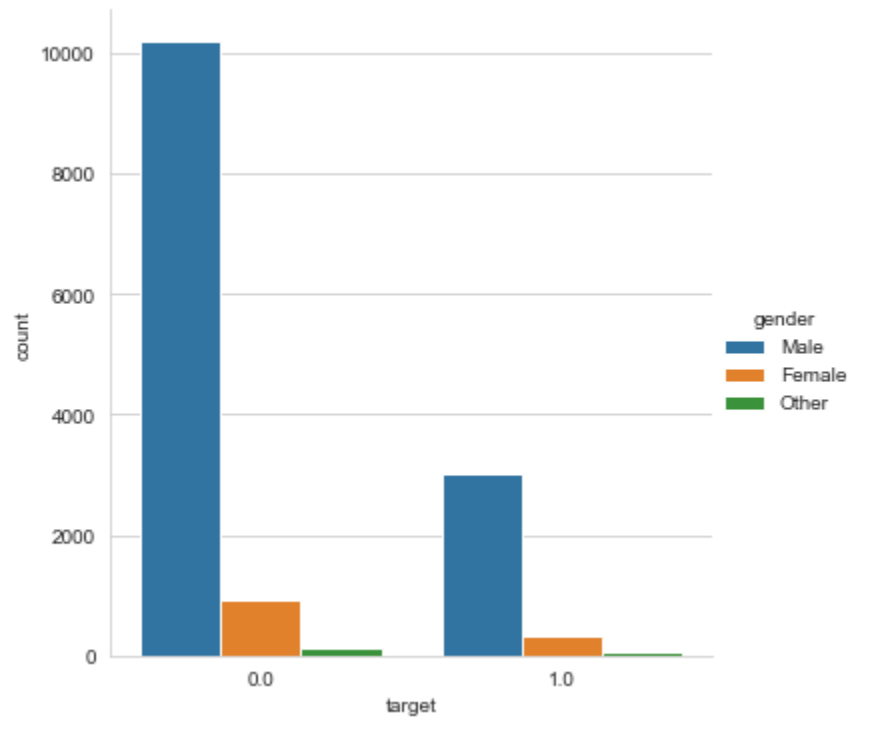
Out[6]:
```
<AxesSubplot:xlabel='target', ylabel='count'>
```



In [7]:
```
sns.catplot('target', data=aug_train, kind='count',hue='gender')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

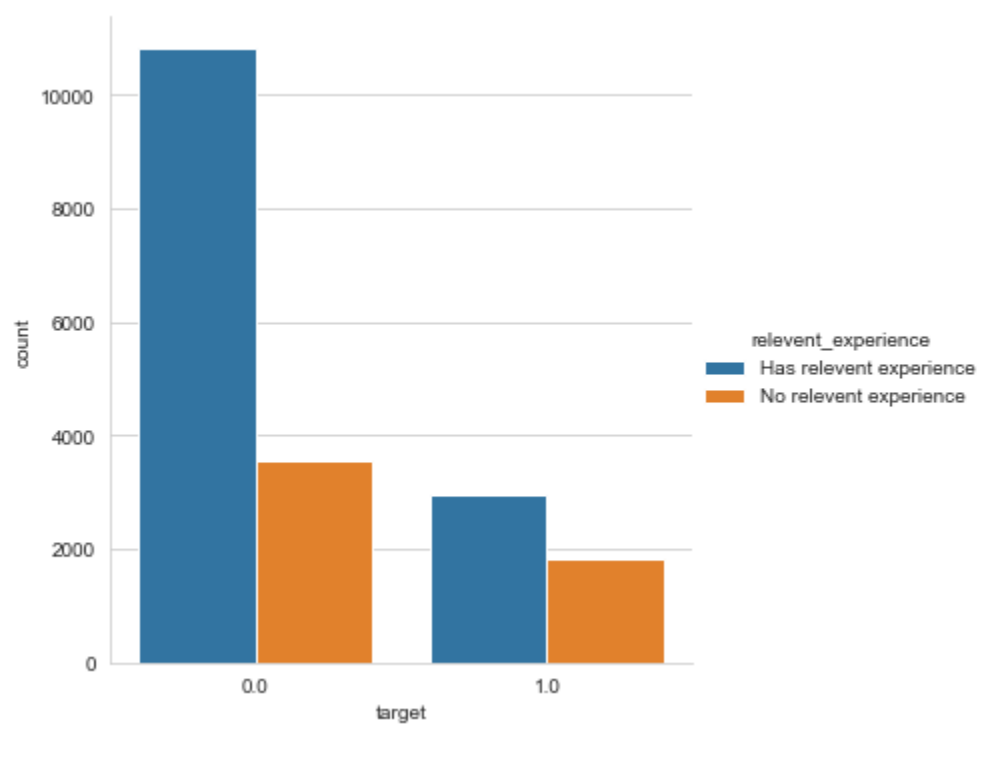
Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x24611e10898>
```



In [8]:
```
sns.catplot('target', data=aug_train, kind='count',hue='relevent_experience')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

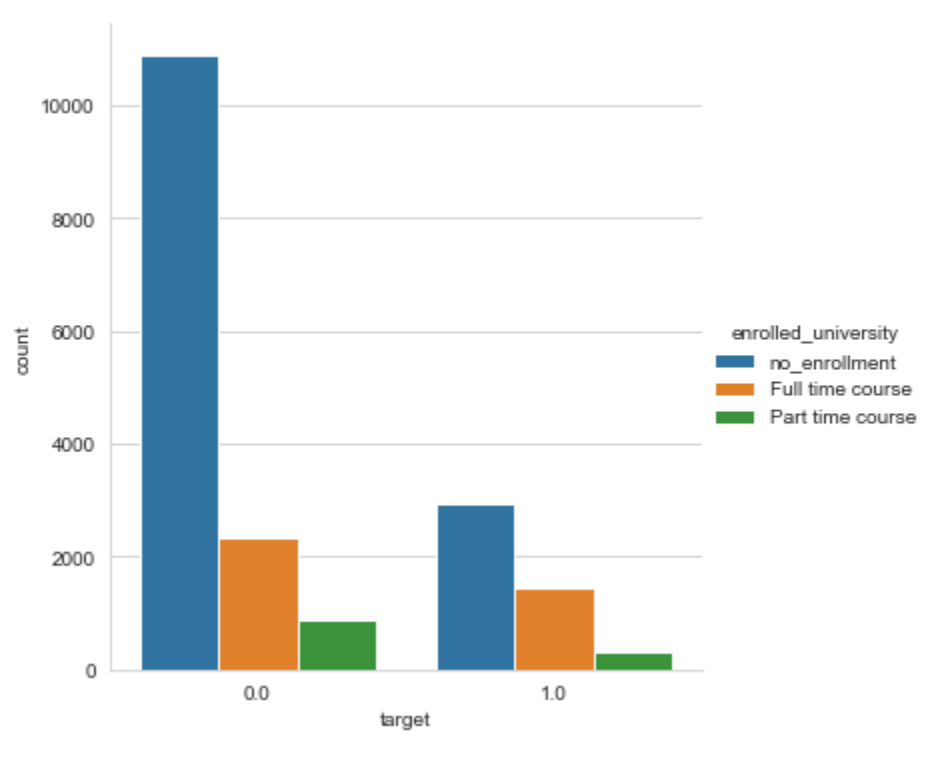
Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x245f9ef2f28>
```



In [9]:
```
sns.catplot('target', data=aug_train, kind='count',hue='enrolled_university')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

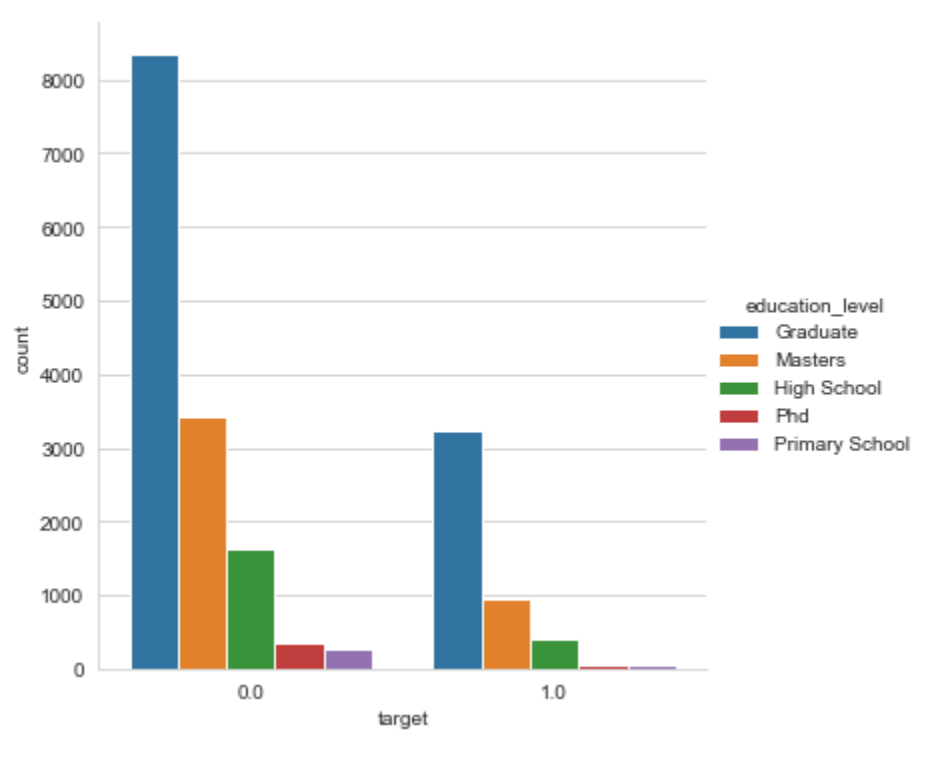
Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x24615ede0b8>
```



In [10]:
```
sns.catplot('target', data=aug_train, kind='count',hue='education_level')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

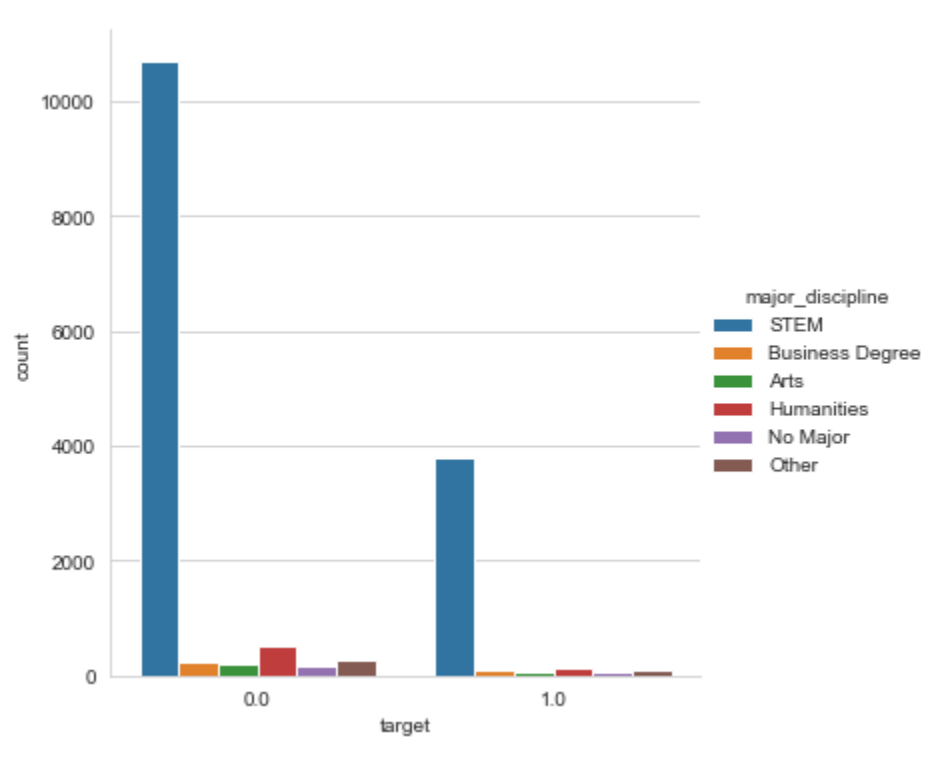
Out[10]:
```
<seaborn.axisgrid.FacetGrid at 0x24615ab0240>
```



In [11]:
```
sns.catplot('target', data=aug_train, kind='count',hue='major_discipline')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

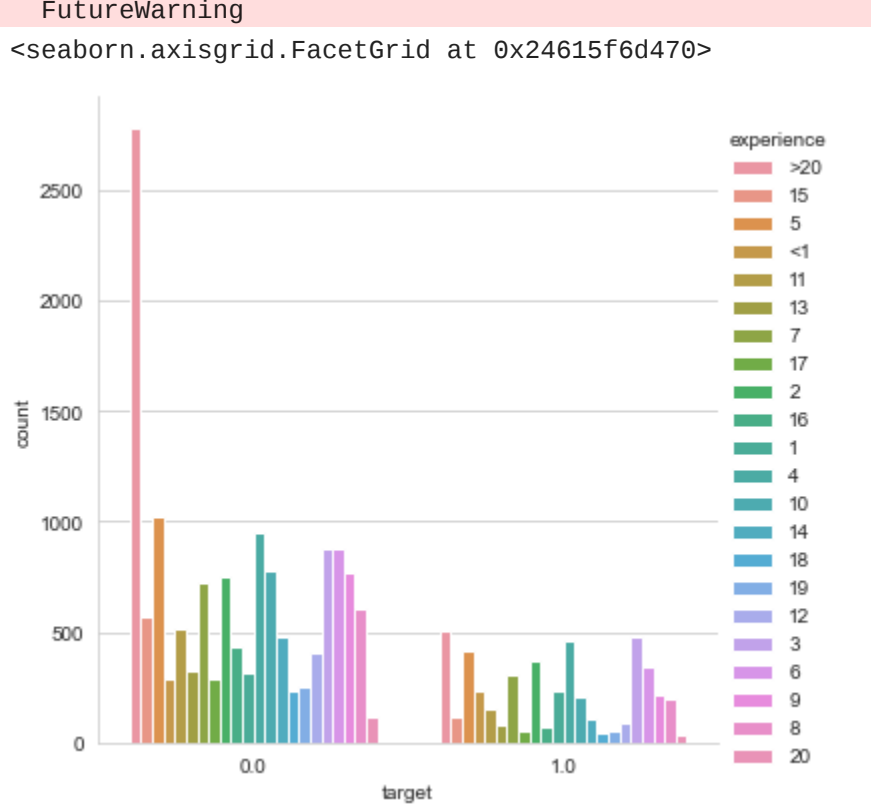
Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x24615ab05c0>
```



In [12]:
```
sns.catplot('target', data=aug_train, kind='count',hue='experience')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

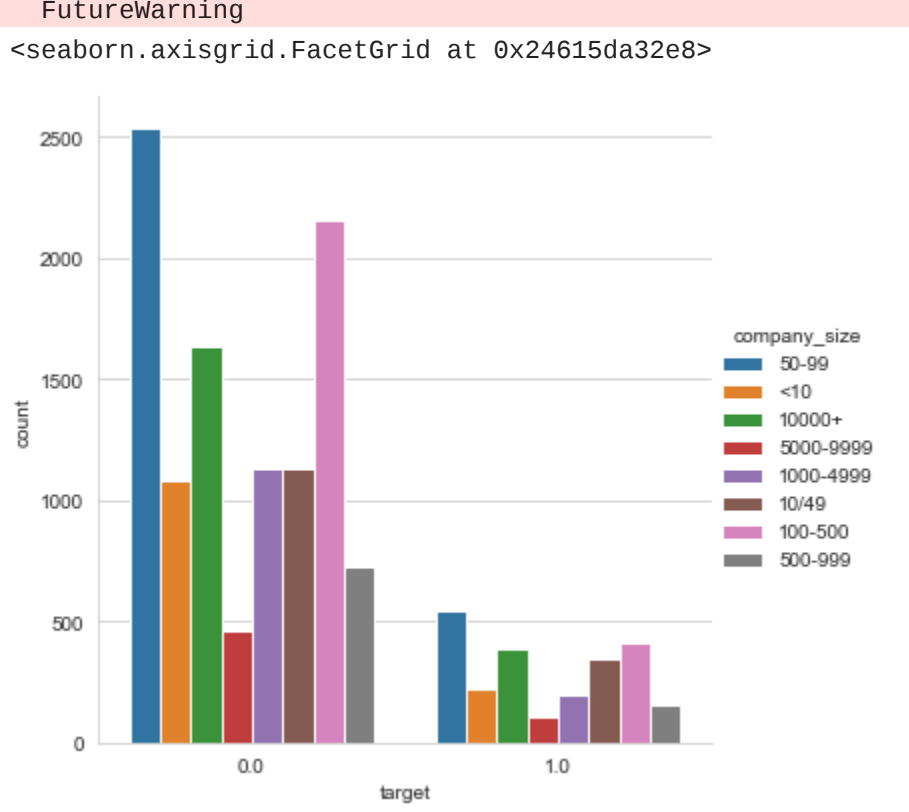
Out[12]:
```
<seaborn.axisgrid.FacetGrid at 0x24615f6d470>
```



In [13]:
```
sns.catplot('target', data=aug_train, kind='count',hue='company_size')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x24615da32e8>
```

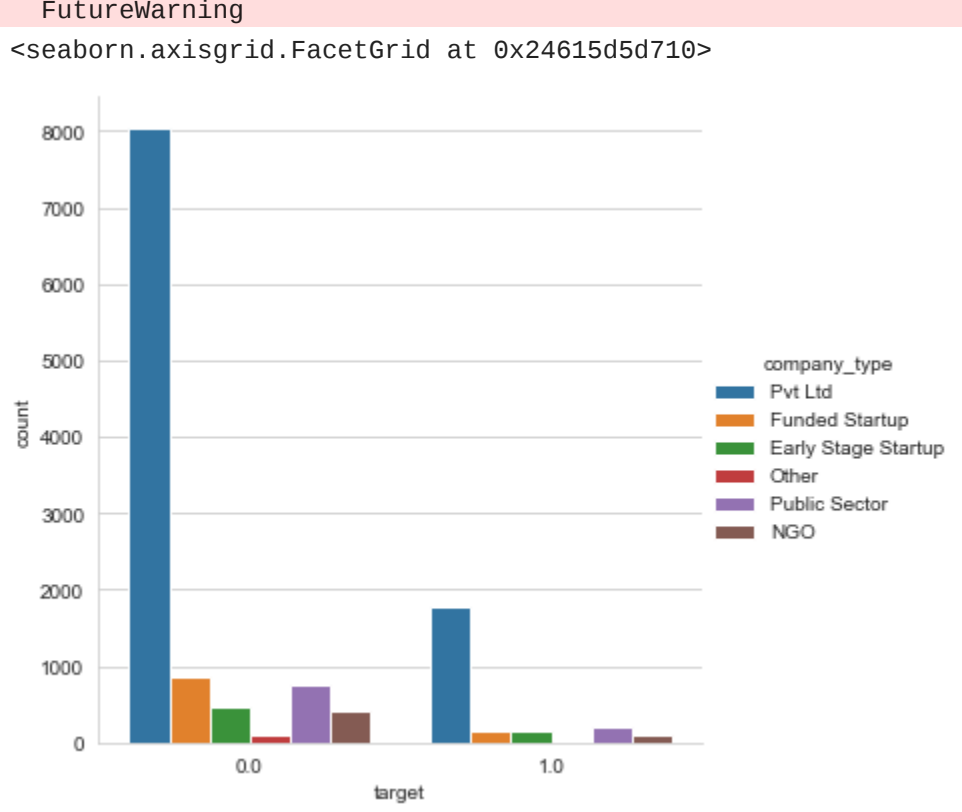


In [14]:
```
aug_train['city'] = aug_train['city'].str.replace(r'\D','')
```

In [15]:
```
sns.catplot('target', data=aug_train, kind='count',hue='company_type')
```

```
c:\users\asus\appdata\local\programs\python\python36\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[15]:
```
<seaborn.axisgrid.FacetGrid at 0x24615d5d710>
```



In [16]:
```
 pd.options.display.float_format = '{:,.0f}'.format
```

In [17]:
```
aug_train = aug_train.fillna(aug_train.mode().iloc[0])
```

In [18]:
```
aug_train.head()
```

Out[18]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	training_hours	target
0	8949	103	1	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20	50-99	Pvt Ltd	1	36	1
1	29725	40	1	Male	No relevent experience	no_enrollment	Graduate	STEM	15	50-99	Pvt Ltd	>4	47	0
2	11561	21	1	Male	No relevent experience	Full time course	Graduate	STEM	5	50-99	Pvt Ltd	never	83	0
3	33241	115	1	Male	No relevent experience	no_enrollment	Graduate	Business Degree	<1	50-99	Pvt Ltd	never	52	1
4	666	162	1	Male	Has relevent experience	no_enrollment	Masters	STEM	>20	50-99	Funded Startup	4	8	0

In [19]:
```
print(aug_train.isnull().sum())
```

```
enrollee_id               0
city                      0
city_development_index    0
gender                    0
relevent_experience       0
enrolled_university       0
education_level           0
major_discipline          0
experience                0
company_size              0
company_type              0
last_new_job              0
training_hours            0
target                    0
dtype: int64
```

In [20]:
```
from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
```

In [21]:
```
obj_df = aug_train.select_dtypes(include=['object']).copy()
int_df = aug_train.select_dtypes(include=['int64']).copy()
float_df = aug_train.select_dtypes(include=['float64']).copy()


le = preprocessing.LabelEncoder()

obj_df_trf = obj_df.astype(str).apply(le.fit_transform)
float_df_trf = float_df.astype(str).apply(le.fit_transform)

df_train = pd.concat([obj_df_trf,float_df_trf,int_df], axis = 1)

df_train.head()
```

Out[21]:

	city	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	city_development_index	target	enrollee_id	training_hours
0	5	1	0	2	0	5	21	4	5	0	85	1	8949	36
1	77	1	1	2	0	5	6	4	5	4	48	0	29725	47
2	64	1	1	0	0	5	15	4	5	5	14	0	11561	83
3	14	1	1	2	0	1	20	4	5	5	52	1	33241	52
4	50	1	0	2	2	5	21	4	1	3	45	0	666	8

In [22]:
```
df_train = df_train.drop(['city_development_index','enrollee_id'], axis=1)
```

In [23]:
```
df_train.head()
```

Out[23]:

	city	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	target	training_hours
0	5	1	0	2	0	5	21	4	5	0	1	36
1	77	1	1	2	0	5	6	4	5	4	0	47
2	64	1	1	0	0	5	15	4	5	5	0	83
3	14	1	1	2	0	1	20	4	5	5	1	52
4	50	1	0	2	2	5	21	4	1	3	0	8

In [24]:
```
df_train = df_train.reindex(columns= ['city', 'gender', 'relevent_experience',
       'enrolled_university', 'education_level', 'major_discipline',
       'experience', 'company_size', 'company_type', 'last_new_job',
       'training_hours', 'target'])
df_train.shape
```

Out[24]:
```
(19158, 12)
```

In [25]:
```
from sklearn.model_selection import train_test_split
X = df_train.iloc[:,0:11].values
y = df_train.iloc [:,11].values
```

APPLYING SMOTE METHOD TO DEAL WITH IMBALANCED DATA

In [26]:
```
from collections import Counter
# summarize class distribution
counter = Counter(y)
print(counter)
```

```
Counter({0: 14381, 1: 4777})
```

In [27]:
```
from imblearn.over_sampling import SMOTE
# transform the dataset
oversample = SMOTE()
X, y = oversample.fit_resample(X, y)
# summarize the new class distribution
counter = Counter(y)
print(counter)
```

```
Counter({1: 14381, 0: 14381})
```

In [28]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 0)
```

After testing many models, Random Forest has the highest accuracy

In [29]:
```
from sklearn.ensemble import RandomForestClassifier
#Random Forest
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, y_train)
Y_pred = random_forest.predict(X_test)
```

In [30]:
```
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, log_loss, roc_auc_score, f1_score, recall_score
```

PERFORMANCE OF OUR MODEL

In [31]:
```
print('ROCAUC score:',roc_auc_score(y_test, Y_pred))
print('Accuracy score:',accuracy_score(y_test, Y_pred))
print('F1 score:',f1_score(y_test, Y_pred))
print('Confusion matrix:' ,confusion_matrix(y_test,Y_pred))
print('Log Loss:' ,log_loss(y_test,Y_pred))
print('Recall:' ,recall_score(y_test,Y_pred))
print(classification_report(y_test,Y_pred))
```

```
ROCAUC score: 0.8094069763225643
Accuracy score: 0.8094907005040848
F1 score: 0.8126495726495727
Confusion matrix: [[2280  573]
 [ 523 2377]]
Log Loss: 6.580037736511317
Recall: 0.8196551724137932
              precision    recall  f1-score   support

           0       0.81      0.80      0.81      2853
           1       0.81      0.82      0.81      2900

    accuracy                           0.81      5753
   macro avg       0.81      0.81      0.81      5753
weighted avg       0.81      0.81      0.81      5753
```

APPLYING THE MODEL TO TEST DATA

In [32]:
```
aug_test['city'] = aug_test['city'].str.replace(r'\D','')
```

In [33]:
```
aug_test = aug_test.fillna(aug_test.mode().iloc[0])
```

In [34]:
```
print(aug_test.isnull().sum())
```

```
enrollee_id               0
city                      0
city_development_index    0
gender                    0
relevent_experience       0
enrolled_university       0
education_level           0
major_discipline          0
experience                0
company_size              0
company_type              0
last_new_job              0
training_hours            0
dtype: int64
```

In [35]:
```
obj_df = aug_test.select_dtypes(include=['object']).copy()
int_df = aug_test.select_dtypes(include=['int64']).copy()
float_df = aug_test.select_dtypes(include=['float64']).copy()


le = preprocessing.LabelEncoder()

obj_df_trf = obj_df.astype(str).apply(le.fit_transform)
float_df_trf = float_df.astype(str).apply(le.fit_transform)

df_test = pd.concat([obj_df_trf,float_df_trf,int_df], axis = 1)

df_test.head()
```

Out[35]:

	city	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	city_development_index	enrollee_id	training_hours
0	69	1	0	0	0	5	19	7	5	0	48	32403	21
1	5	0	0	2	0	5	15	4	5	0	73	9858	98
2	55	1	1	2	1	5	20	4	5	5	14	31806	15
3	22	1	0	2	2	5	2	0	5	0	48	27385	39
4	5	1	0	2	0	5	21	3	5	4	73	27724	72

In [36]:
```
df_test = df_test.drop(['enrollee_id'], axis=1)
```

In [37]:
```
df_test = df_test.reindex(columns= ['city', 'gender', 'relevent_experience',
       'enrolled_university', 'education_level', 'major_discipline',
       'experience', 'company_size', 'company_type', 'last_new_job',
       'training_hours'])
df_test.shape
```

Out[37]:
```
(2129, 11)
```

In [38]:
```
pred = random_forest.predict(df_test)
```

In [39]:
```
pred
```

Out[39]:
```
array([0, 1, 0, ..., 0, 1, 0])
```

In [40]:
```
submission = pd.DataFrame()
submission['enrollee_id']=aug_test['enrollee_id']
submission['target']=pred
submission.to_csv('submission.csv')
```

In [41]:
```
submission.head()
```

Out[41]:

	enrollee_id	target
0	32403	0
1	9858	1
2	31806	0
3	27385	0
4	27724	0