In [260]:
```
import os
os.chdir(r'C:\Users\joyce\OneDrive\Desktop\DATASCIENCE\Python\My Project')
os.getcwd()
```

Out[260]:
```
'C:\\Users\\joyce\\OneDrive\\Desktop\\DATASCIENCE\\Python\\My Project'
```

In [261]:
```
# Importing the numpy and pandas package
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O

!pip install https://github.com/pandas-profiling/pandas-profiling/archive/master.
import pandas_profiling

# Data Visualisation
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import matplotlib.gridspec as gridspec
import warnings
warnings.filterwarnings('ignore')
```

```
Collecting https://github.com/pandas-profiling/pandas-profiling/archive/master.
zip (https://github.com/pandas-profiling/pandas-profiling/archive/master.zip)
  Using cached https://github.com/pandas-profiling/pandas-profiling/archive/mas
ter.zip (https://github.com/pandas-profiling/pandas-profiling/archive/master.zi
p) (34.6 MB)
Requirement already satisfied (use --upgrade to upgrade): pandas-profiling==2.1
2.0 from https://github.com/pandas-profiling/pandas-profiling/archive/master.zi
p (https://github.com/pandas-profiling/pandas-profiling/archive/master.zip) in
 c:\users\joyce\anaconda3\lib\site-packages
Requirement already satisfied: joblib in c:\users\joyce\anaconda3\lib\site-pack
ages (from pandas-profiling==2.12.0) (0.17.0)
Requirement already satisfied: scipy>=1.4.1 in c:\users\joyce\anaconda3\lib\sit
e-packages (from pandas-profiling==2.12.0) (1.5.2)
Requirement already satisfied: pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.25.3 i
n c:\users\joyce\anaconda3\lib\site-packages (from pandas-profiling==2.12.0)
 (1.1.3)
Requirement already satisfied: matplotlib>=3.2.0 in c:\users\joyce\anaconda3\li
b\site-packages (from pandas-profiling==2.12.0) (3.3.2)
Requirement already satisfied: confuse>=1.0.0 in c:\users\joyce\anaconda3\lib\s
ite-packages (from pandas-profiling==2.12.0) (1.4.0)
Requirement already satisfied: jinja2>=2.11.1 in c:\users\joyce\anaconda3\lib\s
ite-packages (from pandas-profiling==2.12.0) (2.11.2)
Requirement already satisfied: visions[type_image_path]==0.6.0 in c:\users\joyc
e\anaconda3\lib\site-packages (from pandas-profiling==2.12.0) (0.6.0)
Requirement already satisfied: numpy>=1.16.0 in c:\users\joyce\anaconda3\lib\si
te-packages (from pandas-profiling==2.12.0) (1.19.2)
Requirement already satisfied: attrs>=19.3.0 in c:\users\joyce\anaconda3\lib\si
te-packages (from pandas-profiling==2.12.0) (20.3.0)
Requirement already satisfied: htmlmin>=0.1.12 in c:\users\joyce\anaconda3\lib
\site-packages (from pandas-profiling==2.12.0) (0.1.12)
Requirement already satisfied: missingno>=0.4.2 in c:\users\joyce\anaconda3\lib
\site-packages (from pandas-profiling==2.12.0) (0.4.2)
Requirement already satisfied: phik>=0.10.0 in c:\users\joyce\anaconda3\lib\sit
e-packages (from pandas-profiling==2.12.0) (0.11.2)
Requirement already satisfied: tangled-up-in-unicode>=0.0.6 in c:\users\joyce\a
naconda3\lib\site-packages (from pandas-profiling==2.12.0) (0.0.7)
Requirement already satisfied: requests>=2.24.0 in c:\users\joyce\anaconda3\lib
\site-packages (from pandas-profiling==2.12.0) (2.24.0)
Requirement already satisfied: tqdm>=4.48.2 in c:\users\joyce\anaconda3\lib\sit
e-packages (from pandas-profiling==2.12.0) (4.50.2)
Requirement already satisfied: seaborn>=0.10.1 in c:\users\joyce\anaconda3\lib
\site-packages (from pandas-profiling==2.12.0) (0.11.0)
```

```
Requirement already satisfied: pytz>=2017.2 in c:\users\joyce\anaconda3\lib\sit
e-packages (from pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.25.3->pandas-profili
ng==2.12.0) (2020.1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\joyce\anacond
a3\lib\site-packages (from pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.25.3->pand
as-profiling==2.12.0) (2.8.1)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\joyce\anaconda3
\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling==2.12.0) (2020.6.2
0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\joyce\anaconda3\li
b\site-packages (from matplotlib>=3.2.0->pandas-profiling==2.12.0) (1.3.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\u
sers\joyce\anaconda3\lib\site-packages (from matplotlib>=3.2.0->pandas-profilin
g==2.12.0) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in c:\users\joyce\anaconda3\lib\si
te-packages (from matplotlib>=3.2.0->pandas-profiling==2.12.0) (8.0.1)
Requirement already satisfied: cycler>=0.10 in c:\users\joyce\anaconda3\lib\sit
e-packages (from matplotlib>=3.2.0->pandas-profiling==2.12.0) (0.10.0)
Requirement already satisfied: pyyaml in c:\users\joyce\anaconda3\lib\site-pack
ages (from confuse>=1.0.0->pandas-profiling==2.12.0) (5.3.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\joyce\anaconda3\lib
\site-packages (from jinja2>=2.11.1->pandas-profiling==2.12.0) (1.1.1)
Requirement already satisfied: networkx>=2.4 in c:\users\joyce\anaconda3\lib\si
te-packages (from visions[type_image_path]==0.6.0->pandas-profiling==2.12.0)
 (2.5)
Requirement already satisfied: imagehash; extra == "type_image_path" in c:\user
s\joyce\anaconda3\lib\site-packages (from visions[type_image_path]==0.6.0->pand
as-profiling==2.12.0) (4.2.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\us
ers\joyce\anaconda3\lib\site-packages (from requests>=2.24.0->pandas-profiling=
=2.12.0) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\joyce\anaconda3\li
b\site-packages (from requests>=2.24.0->pandas-profiling==2.12.0) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\joyce\anaconda3\lib\sit
e-packages (from requests>=2.24.0->pandas-profiling==2.12.0) (2.10)
Requirement already satisfied: six>=1.5 in c:\users\joyce\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7.3->pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.
25.3->pandas-profiling==2.12.0) (1.15.0)
Requirement already satisfied: decorator>=4.3.0 in c:\users\joyce\anaconda3\lib
\site-packages (from networkx>=2.4->visions[type_image_path]==0.6.0->pandas-pro
filing==2.12.0) (4.4.2)
Requirement already satisfied: PyWavelets in c:\users\joyce\anaconda3\lib\site-
packages (from imagehash; extra == "type_image_path"->visions[type_image_path]=
=0.6.0->pandas-profiling==2.12.0) (1.1.1)
Building wheels for collected packages: pandas-profiling
  Building wheel for pandas-profiling (setup.py): started
  Building wheel for pandas-profiling (setup.py): finished with status 'done'
  Created wheel for pandas-profiling: filename=pandas_profiling-2.12.0-py2.py3-
none-any.whl size=243837 sha256=8bbd5e9ef086457c335b0bae1a4566f7122ba5981aea0b4
67c45e2b3833582e1
  Stored in directory: C:\Users\joyce\AppData\Local\Temp\pip-ephem-wheel-cache-
9xo8qei8\wheels\64\b6\85\dfc808b23666a5910371784e349d28818006ff63ed9cfeca59
Successfully built pandas-profiling
```

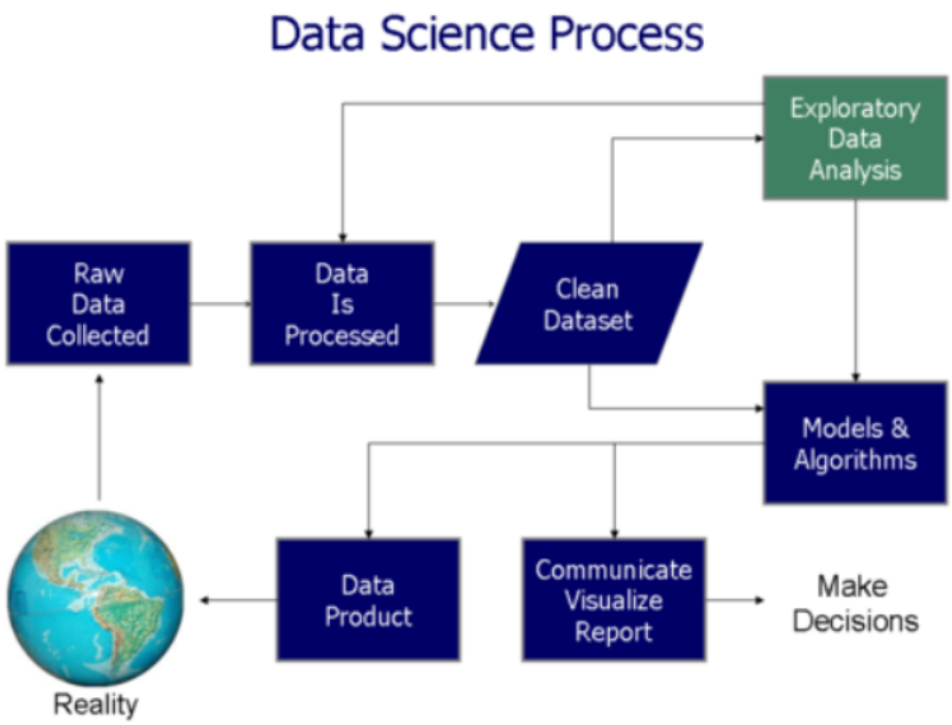

Business Understanding
Data Understanding
Data Preparation
Data
Modeling
Deployment
Evaluation


```
Context and Content
A company which is active in Big Data and Data Science wants to hire data
scientists among people who successfully pass some courses which conduct by the
company. Many people signup for their training. Company wants to know which of
these candidates are really wants to work for the company after training or
looking for a new employment because it helps to reduce the cost and time as
well as the quality of training or planning the courses and categorization of
candidates. Information related to demographics, education, experience are in
hands from candidates signup and enrollment.

This dataset designed to understand the factors that lead a person to leave
current job for HR researches too. By model(s) that uses the current
credentials,demographics,experience data you will predict the probability of a
candidate to look for a new job or will work for the company, as well as
interpreting affected factors on employee decision.
```

loading csv data to dataframe

In [262]:
```
train= pd.read_csv('HR_train.csv',na_values='NA')
test= pd.read_csv('HR_test.csv',na_values='NA')# this a data that we want to pred
```

EDA(Exploratory Data Analysis)



In [263]:
```
#concatenate test and train
train['source']='train'# craeting new column and assign a value ('train') to help
test['source']='test'
df = pd.concat([train,test],ignore_index=True, sort=True)
train.shape , test.shape,df.shape
```

Out[263]:
```
((19158, 15), (2129, 14), (21287, 15))
```

In [264]:
```
#checking  number of obs and columns ,index of columns, name of columns, number c
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21287 entries, 0 to 21286
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   city                    21287 non-null  object 
 1   city_development_index  21287 non-null  float64
 2   company_size            14727 non-null  object 
 3   company_type            14513 non-null  object 
 4   education_level         20775 non-null  object 
 5   enrolled_university     20870 non-null  object 
 6   enrollee_id             21287 non-null  int64  
 7   experience              21217 non-null  object 
 8   gender                  16271 non-null  object 
 9   last_new_job            20824 non-null  object 
 10  major_discipline        18162 non-null  object 
 11  relevent_experience     21287 non-null  object 
 12  source                  21287 non-null  object 
 13  target                  19158 non-null  float64
 14  training_hours          21287 non-null  int64  
dtypes: float64(2), int64(2), object(11)
memory usage: 2.4+ MB
```

Getting familiar with Data

Getting shape of data

In [265]:
```
df.shape
```

Out[265]:
```
(21287, 15)
```

In [266]:
```
#getting number of rows(obs)
df.shape[0]
```

Out[266]:
```
21287
```

In [267]:
```
#getting number of collumns
df.shape[1]
```

Out[267]:
```
15
```

Checking the head of the dataset

In [268]:
```
# Checking the head of the dataset
df.head()
```

Out[268]:

	city	city_development_index	company_size	company_type	education_level	enrolled_un
0	city_103	0.920	NaN	NaN	Graduate	no_enr
1	city_40	0.776	50-99	Pvt Ltd	Graduate	no_enr
2	city_21	0.624	NaN	NaN	Graduate	Full time
3	city_115	0.789	NaN	Pvt Ltd	Graduate	
4	city_162	0.767	50-99	Funded Startup	Masters	no_enr

In [269]:
```
df.tail()
```

Out[269]:

	city	city_development_index	company_size	company_type	education_level	enrolled_u
21282	city_103	0.920	NaN	Public Sector	Graduate	no_e
21283	city_136	0.897	NaN	NaN	Masters	no_e
21284	city_100	0.887	NaN	Pvt Ltd	Primary School	no_e
21285	city_102	0.804	100-500	Public Sector	High School	Full tim
21286	city_102	0.804	10000+	Pvt Ltd	Masters	no_e

Handling Duplicate Data

```
In real world you ara not allowd to remove any obsevation that belongs to test
(future) data set,
because we have to predict for each observation of tast data set. that's why I
will just remove duplicate data from
train data set.
```

In [270]:
```
#drop duplicate obs from train data set
train=train.drop_duplicates()
```

In [271]:
```
#combining train and test data set to make a df_nodup data set
train['source']='train'
test['source']='test'
df_nodup = pd.concat([train,test],ignore_index=True, sort=True)
print(df.shape,df_nodup.shape,'\n Number of duplicate data : ',df.shape[0]-df_noc
```

```
(21287, 15) (21287, 15)
 Number of duplicate data :  0
```

In [272]:
```
#replace df with df_nodup
df=df_nodup
```

In [273]:
```
import plotly as py
import plotly.graph_objs as go
import plotly.express as px
from plotly.offline import init_notebook_mode
init_notebook_mode(connected = True)
import seaborn as sns

import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import classification_report, confusion_matrix, roc_curve, a
from sklearn.metrics import roc_auc_score, precision_score, recall_score, f1_scor

from sklearn.model_selection import train_test_split, cross_val_score

from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
```

In [274]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\joyce\anaconda3\lib\site-pack
ages (4.14.3)
Requirement already satisfied: six in c:\users\joyce\anaconda3\lib\site-package
s (from plotly) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in c:\users\joyce\anaconda3\lib
\site-packages (from plotly) (1.3.3)
Note: you may need to restart the kernel to use updated packages.
```

In [275]:
```
from pandas_profiling import ProfileReport
```

```
pandas_profiling extends the pandas DataFrame with df.profile_report() for
quick data analysis.

For each column the following statistics - if relevant for the column type -
are presented in an interactive HTML report:

Type inference: detect the types of columns in a dataframe. Essentials: type,
unique values, missing values Quantile statistics like minimum value, Q1,
median, Q3, maximum, range, interquartile range Descriptive statistics like
mean, mode, standard deviation, sum, median absolute deviation, coefficient of
variation, kurtosis, skewness Most frequent values Histogram Correlations
highlighting of highly correlated variables, Spearman, Pearson and Kendall
matrices Missing values matrix, count, heatmap and dendrogram of missing values
Text analysis learn about categories (Uppercase, Space), scripts (Latin,
Cyrillic) and blocks (ASCII) of text data. File and Image analysis extract file
sizes, creation dates and dimensions and scan for truncated images or those
containing EXIF information.
```

In [276]:
```
pandas_profiling.ProfileReport(df)
```

Summarize dataset: 100% 28/28 [00:15<00:00, 1.84it/s, Completed]

Generate report structure: 100% 1/1 [00:07<00:00, 7.78s/it]

Render HTML: 100% 1/1 [00:01<00:00, 1.39s/it]

Most occurring characters

Value	Count	Frequency (%)
c	21287	13.3%
i	21287	13.3%
t	21287	13.3%
y	21287	13.3%
_	21287	13.3%
1	19152	12.0%
0	8004	5.0%
3	7052	4.4%
6	5219	3.3%
2	4533	2.8%
Other values (5)	9490	5.9%

Most occurring categories

Value	Count	Frequency (%)
Lowercase Letter	85148	53.3%
Decimal Number	53450	33.4%
Connector Punctuation	21287	13.3%

Out[276]:

Preparing data for EDA

Missing Values

In [277]:
```
# counting missing values
df.apply(lambda x: sum(x.isnull()))
```

Out[277]:
```
city                        0
city_development_index      0
company_size             6560
company_type             6774
education_level           512
enrolled_university       417
enrollee_id                 0
experience                 70
gender                   5016
last_new_job              463
major_discipline         3125
relevent_experience         0
source                      0
target                   2129
training_hours              0
dtype: int64
```

In [278]:
```
#calculatin no. of missing values for each column and it's percentage
def percentage_of_miss():
  df1=df[df.columns[df.isnull().sum()>=1]] # I did slicing by condition( I get s
  total_miss = df1.isnull().sum().sort_values(ascending=False)
  percent_miss = ((df1.isnull().sum()/df1.isnull().count())*100).sort_values(asce
  missing_data = pd.concat([total_miss, percent_miss], axis=1, keys=['Number of M
  return(missing_data)
```

In [279]:
```
percentage_of_miss()
```

Out[279]:

	Number of Missing	Percentage
company_type	6774	31.822239
company_size	6560	30.816931
gender	5016	23.563677
major_discipline	3125	14.680321
target	2129	10.001409
education_level	512	2.405224
last_new_job	463	2.175036
enrolled_university	417	1.958942
experience	70	0.328839

```
The missing values in this data are not the ones that can be easily to impute,
because if you restore incorrectly, you may actually see non-existent
correlations and, in general, the data logic may be lost. Therefore, the EDA
will be performed on all available data, and for modeling, all rows with
missing values will be deleted.
```

In [280]:
```
#df = df.drop(['enrollee_id', 'city'], axis = 1)
```

In [281]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21287 entries, 0 to 21286
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   city                    21287 non-null  object 
 1   city_development_index  21287 non-null  float64
 2   company_size            14727 non-null  object 
 3   company_type            14513 non-null  object 
 4   education_level         20775 non-null  object 
 5   enrolled_university     20870 non-null  object 
 6   enrollee_id             21287 non-null  int64  
 7   experience              21217 non-null  object 
 8   gender                  16271 non-null  object 
 9   last_new_job            20824 non-null  object 
 10  major_discipline        18162 non-null  object 
 11  relevent_experience     21287 non-null  object 
 12  source                  21287 non-null  object 
 13  target                  19158 non-null  float64
 14  training_hours          21287 non-null  int64  
dtypes: float64(2), int64(2), object(11)
memory usage: 2.4+ MB
```

In [282]:
```
df['company_size'].unique()
```

Out[282]:
```
array([nan, '50-99', '<10', '10000+', '5000-9999', '1000-4999', '10/49',
       '100-500', '500-999'], dtype=object)
```

In [283]:
```
for i in range(len(df.index)):
    if df['company_size'][i] == '10/49':
        df['company_size'][i] = '10-49'
```

In [284]:
```
df['experience'].unique()
```

Out[284]:
```
array(['>20', '15', '5', '<1', '11', '13', '7', '17', '2', '16', '1', '4',
       '10', '14', '18', '19', '12', '3', '6', '9', '8', '20', nan],
      dtype=object)
```

In [285]:
```
for i in range(len(df.index)):
    if df['experience'][i] == '>20':
        df['experience'][i] = '21'
    elif df['experience'][i] == '<1':
        df['experience'][i] = '0'
```

In [286]:
```
df['last_new_job'].unique()
```

Out[286]:
```
array(['1', '>4', 'never', '4', '3', '2', nan], dtype=object)
```

In [287]:
```
    if df['last_new_job'][i] == '>4':
        df['last_new_job'][i] = '5'
    elif df['last_new_job'][i] == 'never':
        df['last_new_job'][i] = '0'
```

In [288]:
```
pip install catboost
```

```
Requirement already satisfied: catboost in c:\users\joyce\anaconda3\lib\site-pa
ckages (0.25)
Requirement already satisfied: matplotlib in c:\users\joyce\anaconda3\lib\site-
packages (from catboost) (3.3.2)
Note: you may need to restart the kernel to use updated packages.Requirement al
ready satisfied: six in c:\users\joyce\anaconda3\lib\site-packages (from catboo
st) (1.15.0)
Requirement already satisfied: plotly in c:\users\joyce\anaconda3\lib\site-pack
ages (from catboost) (4.14.3)
Requirement already satisfied: graphviz in c:\users\joyce\anaconda3\lib\site-pa
ckages (from catboost) (0.16)
Requirement already satisfied: numpy>=1.16.0 in c:\users\joyce\anaconda3\lib\si
te-packages (from catboost) (1.19.2)
Requirement already satisfied: scipy in c:\users\joyce\anaconda3\lib\site-packa
ges (from catboost) (1.5.2)
Requirement already satisfied: pandas>=0.24.0 in c:\users\joyce\anaconda3\lib\s
ite-packages (from catboost) (1.1.3)
Requirement already satisfied: cycler>=0.10 in c:\users\joyce\anaconda3\lib\sit
e-packages (from matplotlib->catboost) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\u
sers\joyce\anaconda3\lib\site-packages (from matplotlib->catboost) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\joyce\anaconda3
\lib\site-packages (from matplotlib->catboost) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\joyce\anaconda3\li
b\site-packages (from matplotlib->catboost) (1.3.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\joyce\anaconda3\lib\si
te-packages (from matplotlib->catboost) (8.0.1)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\joyce\anaconda3
\lib\site-packages (from matplotlib->catboost) (2020.6.20)
Requirement already satisfied: retrying>=1.3.3 in c:\users\joyce\anaconda3\lib
\site-packages (from plotly->catboost) (1.3.3)
Requirement already satisfied: pytz>=2017.2 in c:\users\joyce\anaconda3\lib\sit
e-packages (from pandas>=0.24.0->catboost) (2020.1)
```

In [289]:
```
pip install lightgbm
```

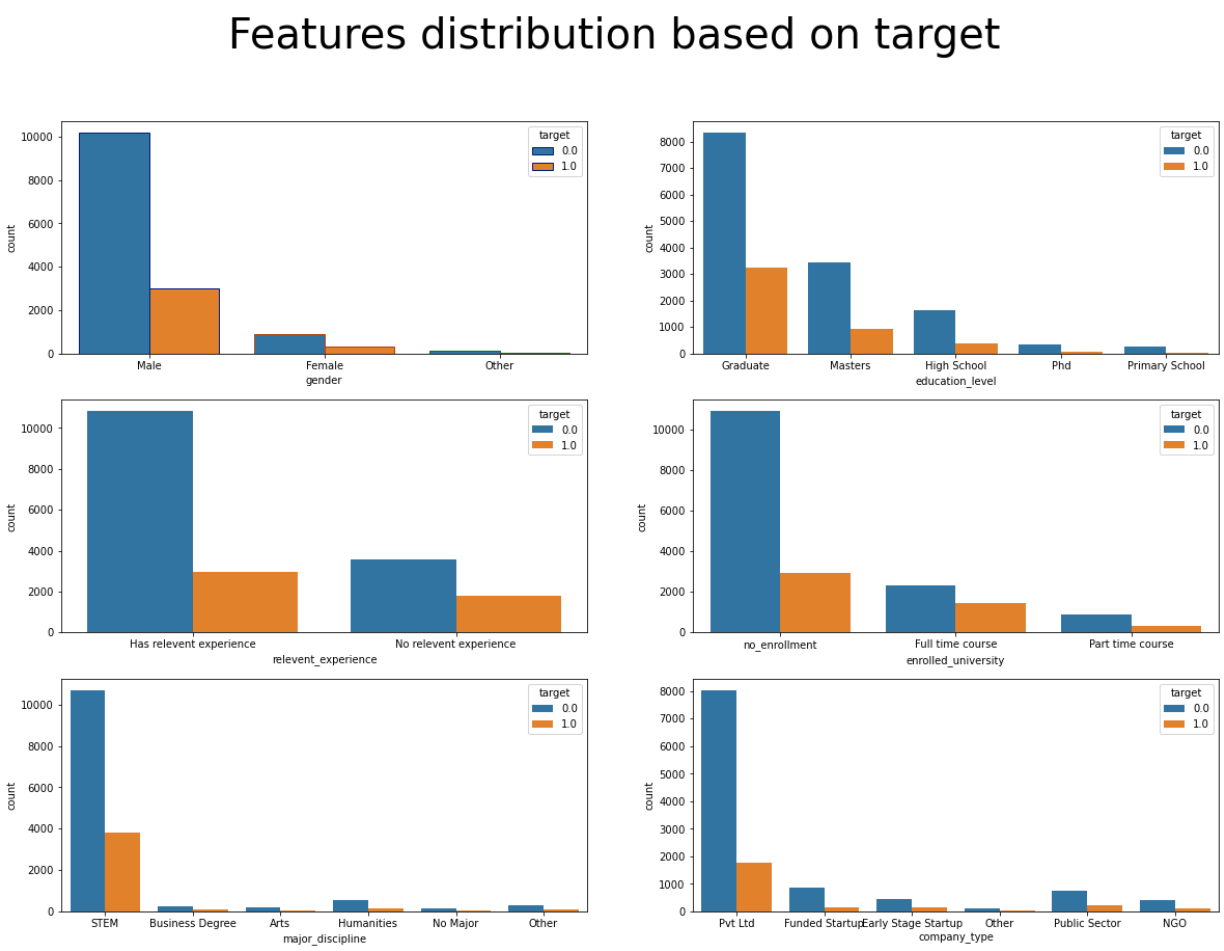
```
Requirement already satisfied: lightgbm in c:\users\joyce\anaconda3\lib\site-pa
ckages (3.2.0)
Requirement already satisfied: numpy in c:\users\joyce\anaconda3\lib\site-packa
ges (from lightgbm) (1.19.2)
Requirement already satisfied: wheel in c:\users\joyce\anaconda3\lib\site-packa
ges (from lightgbm) (0.35.1)
Requirement already satisfied: scipy in c:\users\joyce\anaconda3\lib\site-packa
ges (from lightgbm) (1.5.2)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\joyce\anaconda3
\lib\site-packages (from lightgbm) (0.23.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\joyce\anaconda3
\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\users\joyce\anaconda3\lib\sit
e-packages (from scikit-learn!=0.22.0->lightgbm) (0.17.0)
Note: you may need to restart the kernel to use updated packages.
```

In [290]:
```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\joyce\anaconda3\lib\site-pac
kages (1.3.3)
Requirement already satisfied: scipy in c:\users\joyce\anaconda3\lib\site-packa
ges (from xgboost) (1.5.2)
Requirement already satisfied: numpy in c:\users\joyce\anaconda3\lib\site-packa
ges (from xgboost) (1.19.2)
Note: you may need to restart the kernel to use updated packages.
```

Q1.How all the categorical features effecting in target variable.

In [291]:
```
fig_dims = (20, 14)
fig, ax =plt.subplots(3,2,figsize = fig_dims)
sns.countplot(x = train['gender'],hue = train['target'], ax=ax[0,0], edgecolor=sr
sns.countplot(train['education_level'],hue = train['target'], ax=ax[0,1])
sns.countplot(x = train['relevent_experience'],hue = train['target'], ax=ax[1,0])
sns.countplot(train['enrolled_university'],hue = train['target'], ax=ax[1,1])
sns.countplot(x = train['major_discipline'],hue = train['target'], ax=ax[2,0])
sns.countplot(x = train['company_type'],hue = train['target'], ax=ax[2,1])
fig.suptitle('Features distribution based on target ',fontsize=40)
fig.show()
```



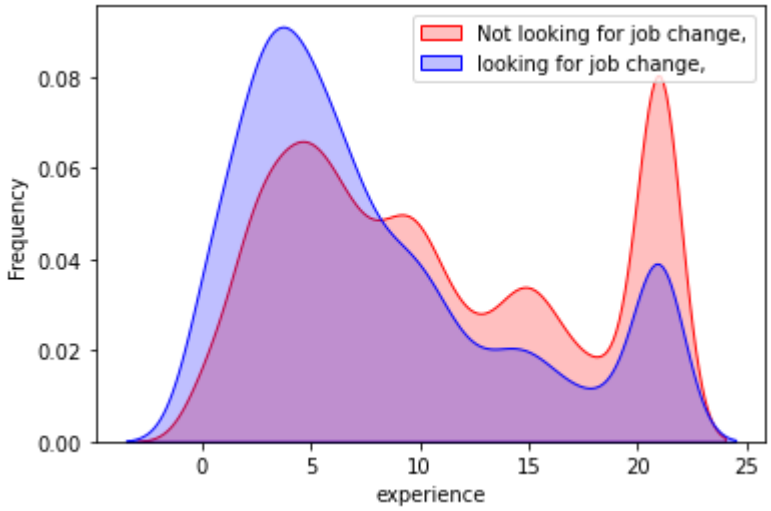
Though something to see is even people from public sector are also getting interest in Data Science.

Peopler are from Arts background are completly not interested in switching job.

Q2. Analysis of experience with target

In [292]:
```
df = df[~df['experience'].isnull()]
df['experience'] = df['experience'].astype(int)
```

In [293]:
```
g = sns.kdeplot(df['experience'][(df["target"] == 0) & (df['experience'].notnull(
g = sns.kdeplot(df['experience'][(df["target"] == 1) & (df['experience'].notnull(
g.set_xlabel('experience')
g.set_ylabel("Frequency")
g = g.legend(["Not looking for job change,","looking for job change,"])
```



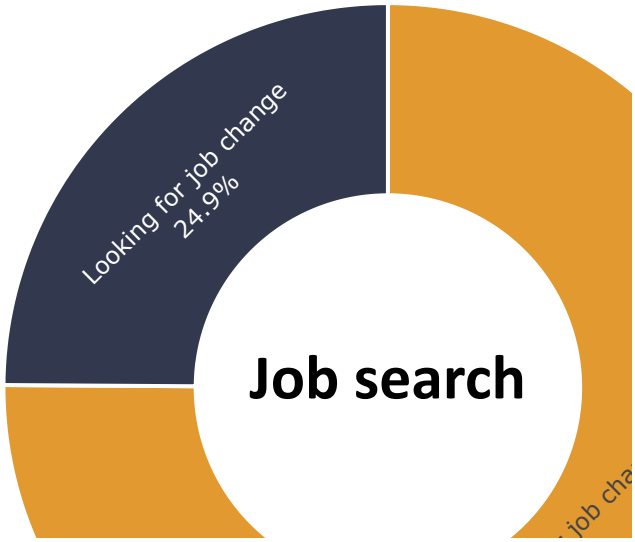
Make Obserbations.

People ranging experience from 1 to 10 years are most likely to change.
People having experience of around 20 years are not looking to change the job.

In [294]:
```
retarget = {0.0: 'Not looking for job change',
            1.0: 'Looking for job change'}
df['target'] = df['target'].map(retarget)
```

Q3. Distribution of Target

In [295]:
```
target = df.groupby('target').agg({'target': 'count'}).rename(columns = {'target'

fig = px.pie(target, values = 'count', names = 'target')
fig.update_traces(textposition = 'inside',
                  textinfo = 'percent + label',
                  hole = 0.5,
                  marker = dict(colors = ['#32384D','#E29930'], line = dict(color

fig.update_layout(title_text = 'Job search', title_x = 0.5, title_y = 0.53, title
                  showlegend = False)

fig.show()
```



Q4.Chi-squared test of independence- find relation between relevant_experience and Target

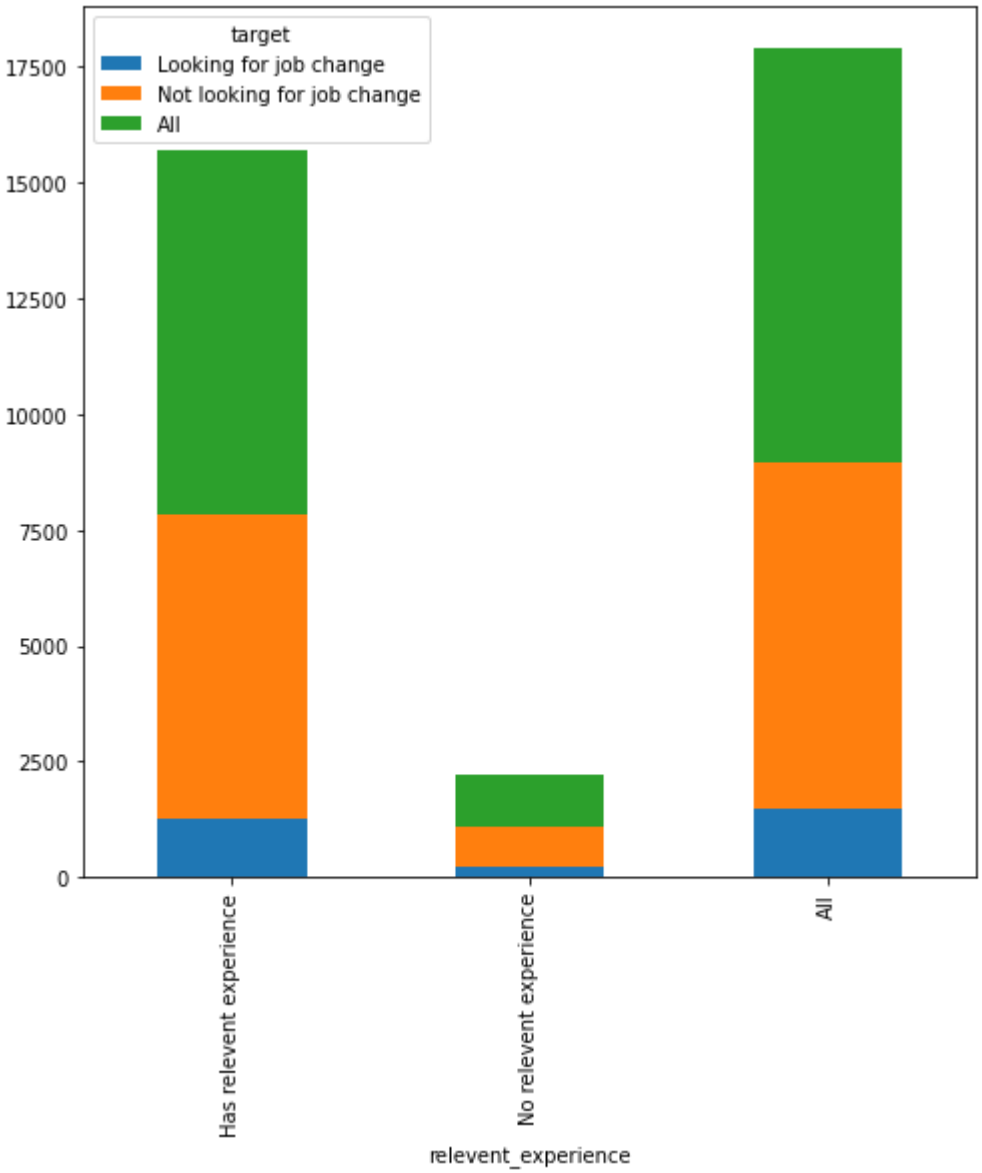
In [296]:
```
contingency_table1 = pd.crosstab(df['relevent_experience'],df['target'],margins =
contingency_table1
```

Out[296]:

target	Looking for job change	Not looking for job change	All
relevent_experience			
Has relevent experience	2945	10803	13748
No relevent experience	1809	3536	5345
All	4754	14339	19093

In [188]:
```
contingency_table1.plot(kind="bar",
                        figsize=(8,8),
                        stacked=True)
```

Out[188]: <AxesSubplot:xlabel='relevent_experience'>



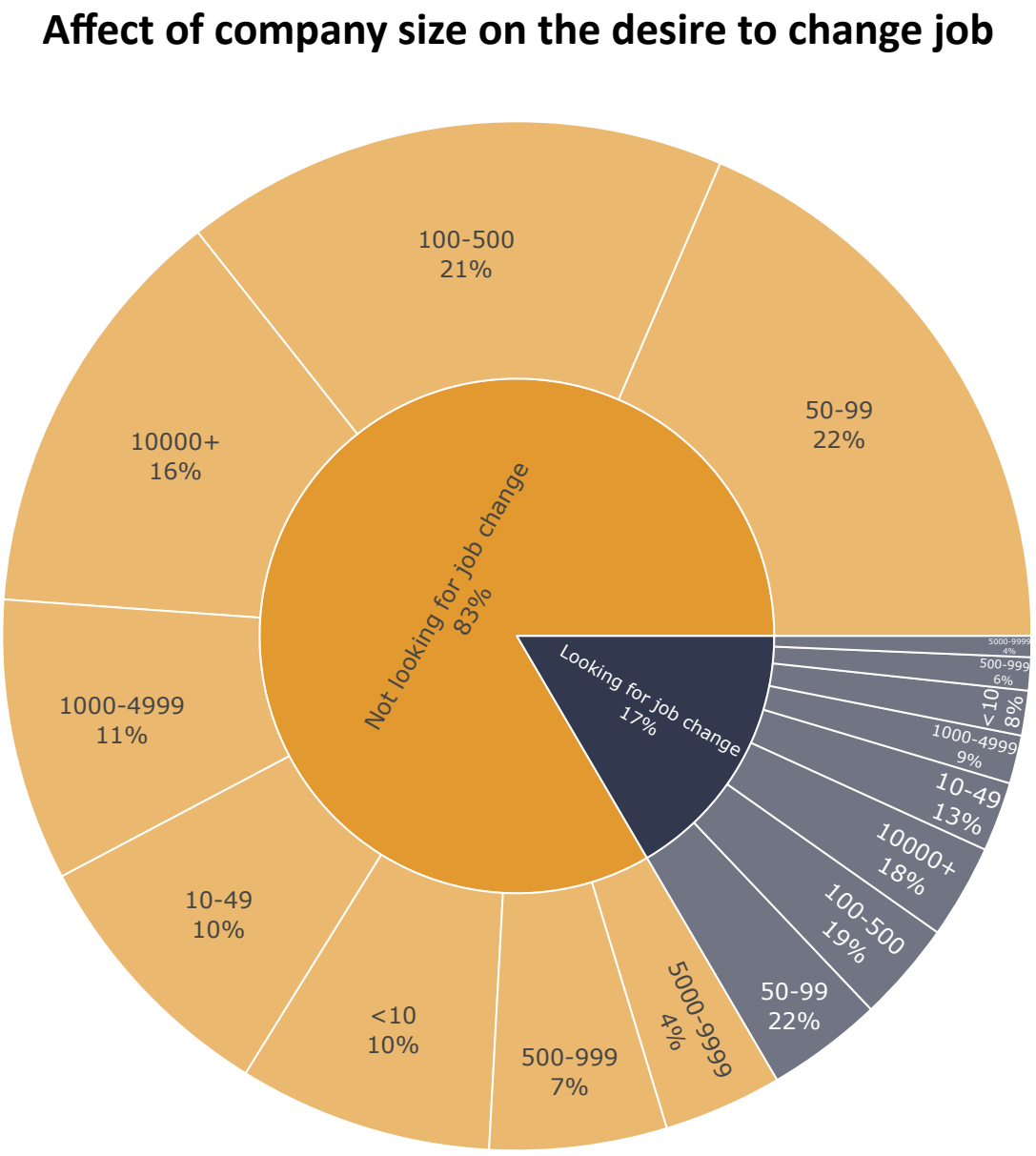
In [297]:
```
from scipy.stats import chi2_contingency
def chi_square(c1,c2):
    chi_2, p_val, dof, exp_val = chi2_contingency(pd.crosstab(df[c1],df[c2],margi

    print(exp_val)
    print('\nChi-square is : %f'%chi_2, '\n\np_value is : %f'%p_val, '\n\ndegree

    if p_val < 0.05:# consider significan level is 5%
        print("\nThere is some correlation between the two variables at significa
    else:
        print("\nThere is no correlation between the two variables")
```

In [298]:
```
chi_square("relevent_experience",'target')
```

```
[[ 3423.13895145 10324.86104855]
 [ 1330.86104855  4014.13895145]]

Chi-square is : 316.998141

p_value is : 0.000000

degree of freedom is : 1

There is some correlation between the two variables at significance level 0.05
```

Q5. How company size is affecting Target

In [143]:
```
cs = df.groupby(['target', 'company_size']).agg({'target': 'count'}).rename(colum

fig = px.sunburst(cs, path = ['target', 'company_size'], values = 'count', color
                  color_discrete_map = {'Looking for job change': '#32384D', 'Not
                  width = 700, height = 700)

fig.update_layout(annotations = [dict(text = 'Affect of company size on the desir
                                      x = 0.5, y = 1.1, font_size = 24, showarrow
                                      font_family = 'Calibri Black',
                                      font_color = 'black')])

fig.update_traces(textinfo = 'label + percent parent')

fig.show()
```

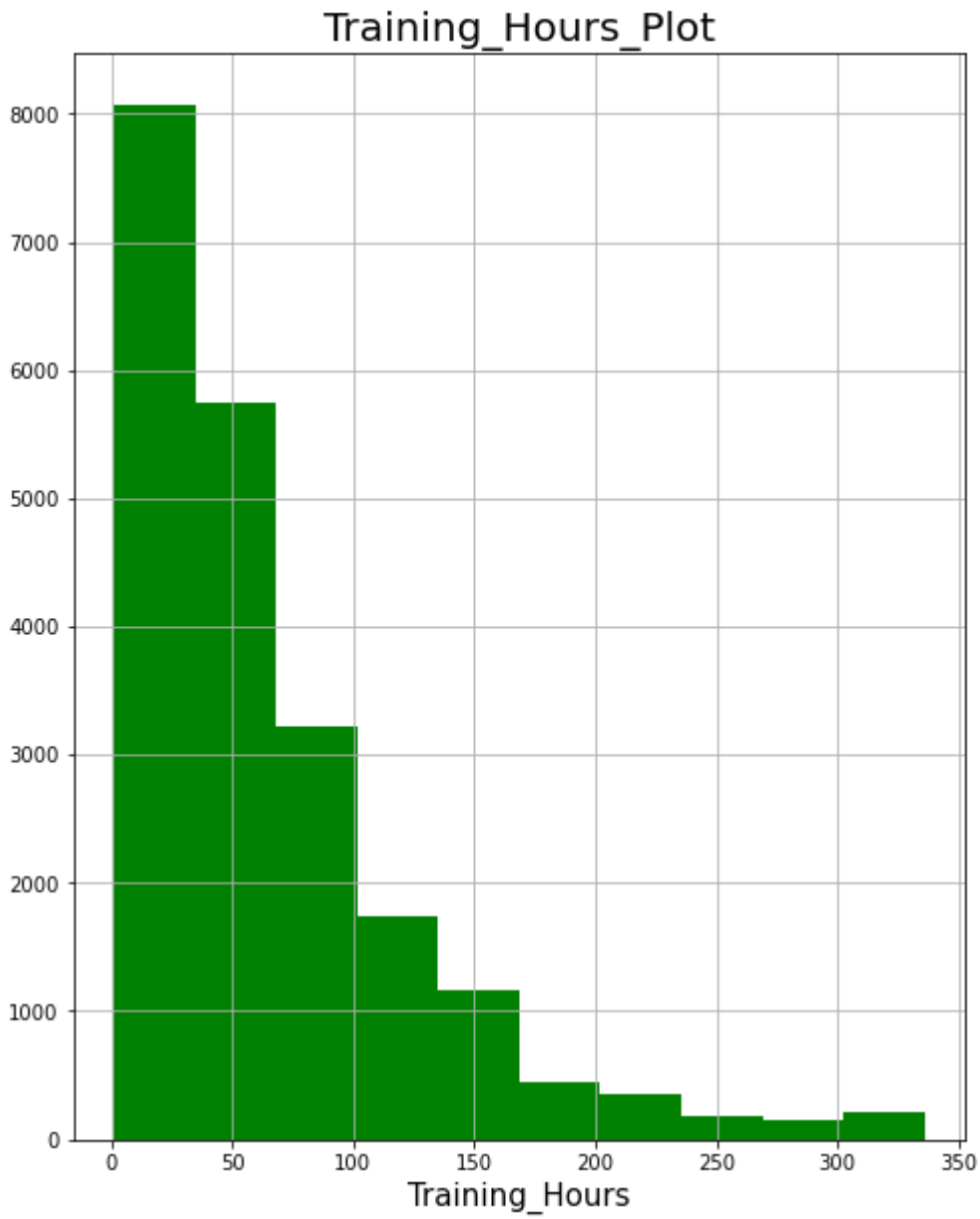


In [ ]:
```
83% dont want to leave the company and
```

Q6. Training hours Analysis

In [135]:
```
#Plot of Training_Hours
print('Fig 1.0')
plt.figure(figsize=(8,10))
plt.title('Training_Hours_Plot',size=20)
plt.xlabel('Training_Hours',size=15)
plt.hist(df.training_hours,color='g')
plt.grid('True')
plt.show()
```

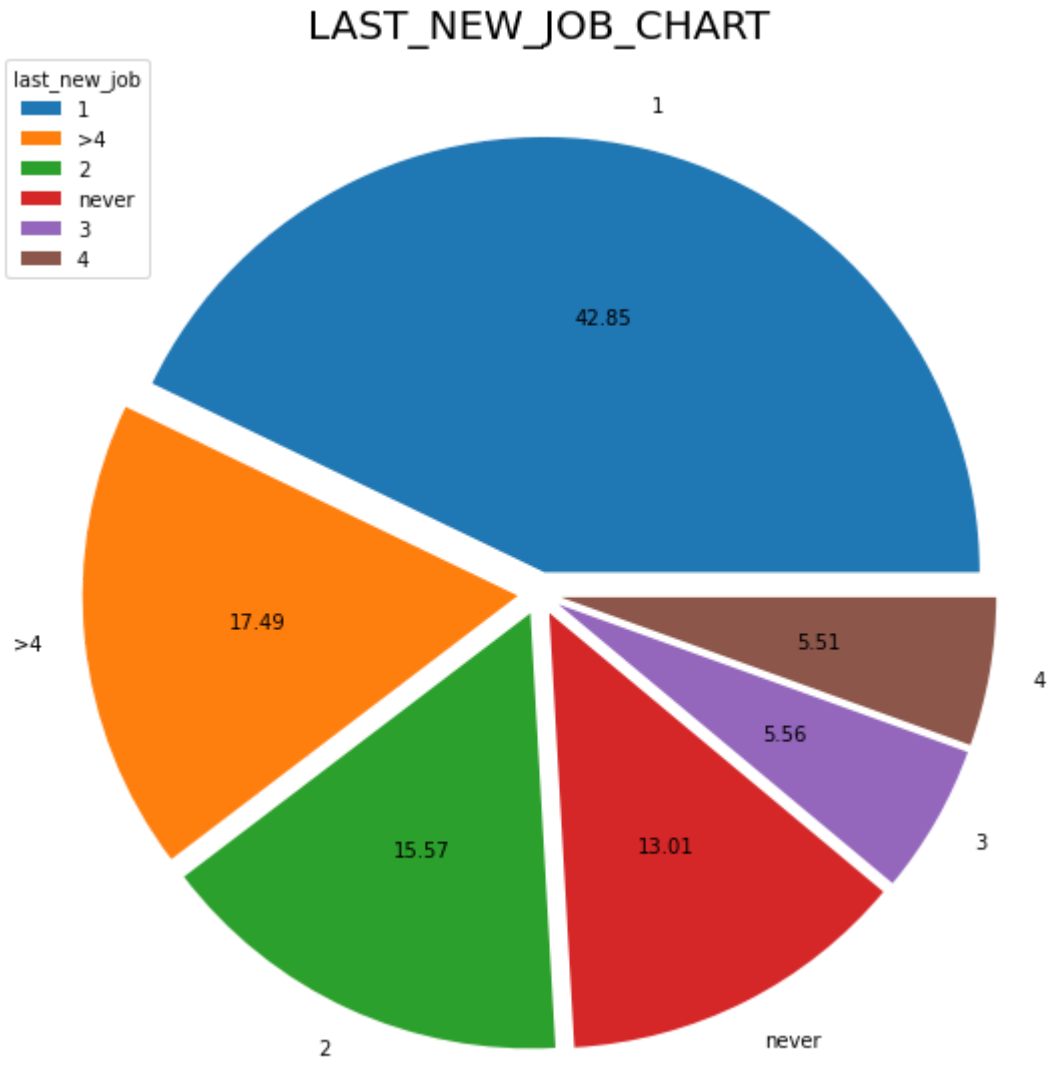
Fig 1.0



Conclusion-Majority of Candidates as can be seen from fig spent less than 100 Hours on Training

Q7. The distribution of the number of jobs previously held by candidates

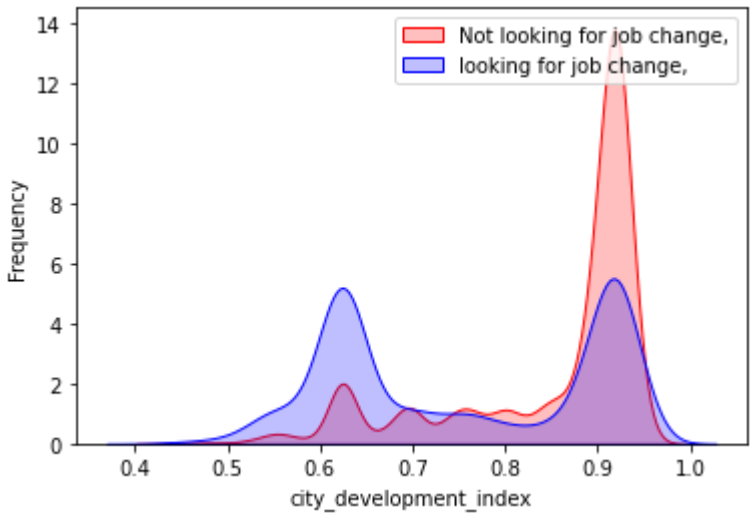
In [134]:
```
#Chart of New Jobs

def piechart(col,x):
    " piechart(col,size);\n 'col' represents the column to be plotted it shld be
    plt.figure(figsize=x)
    plt.title(col.upper()+'_CHART',size=20)
    plt.pie(x=df[col].value_counts().values,explode=[0.05 for x in range(len(df[c
    plt.legend(title=col,loc='best')
    plt.show()
piechart('last_new_job',(10,10))
```



Above is a plot of the distribution of the number of jobs previously held by candidates, Most Candidates have had just 1(One) previous job before applying to the company this represents 42% of the total candidates while candidates who have had more than 4 jobs represents 17% of the total candidates, some 13% of candidates have never had a previous job. It will be interesting to see if work experience influences candidates decision to accept an offer at the company

Q8. The distribution of city_development_index with target

In [236]:
```
g = sns.kdeplot(train['city_development_index'][(train["target"] == 0) & (train['
g = sns.kdeplot(train['city_development_index'][(train["target"] == 1) & (train['
g.set_xlabel('city_development_index')
g.set_ylabel("Frequency")
g = g.legend(["Not looking for job change,","looking for job change,"])
```



conclusion: The CDI (City Development Index) has a big role in the desire to change job: more than half of the specialists with a low CDI are looking for a new job - in cities with a high CDI, which is not strange, the situation is the opposite, more than half of the specialists are not interested in finding a new job.

Q9. Analysis of Company type with Target

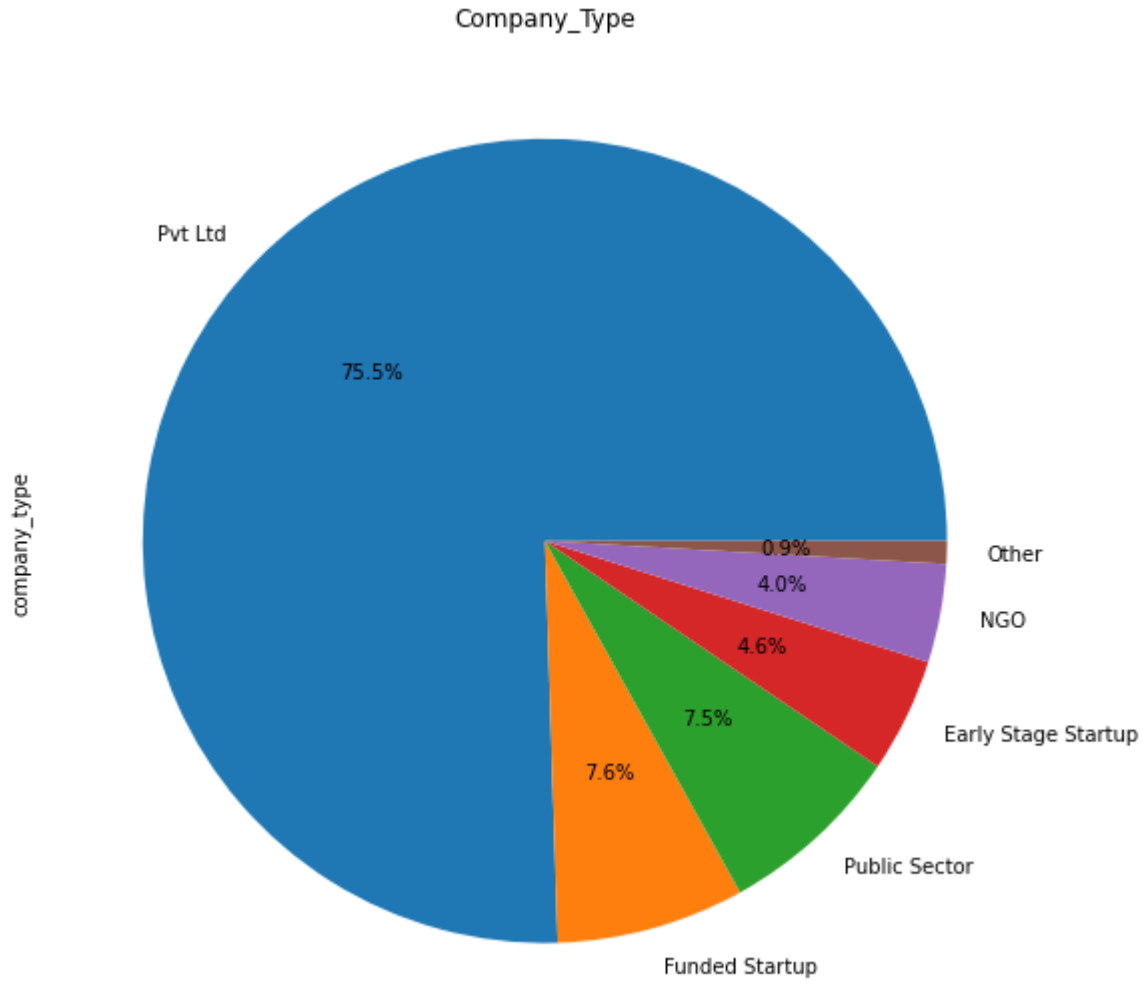
In [248]:
```
retarget2 = {'Not looking for job change': 0, 'Looking for job change': 1}
```

In [240]:
```
df[df["target"] == 1]['company_type'].value_counts()[1:]
```

Out[240]:
```
Series([], Name: company_type, dtype: int64)
```

In [241]:
```
company = df["company_type"].value_counts()
company.plot.pie(figsize = (9,12) , autopct='%1.1f%%' , title = "Company_Type")
```

Out[241]: <AxesSubplot:title={'center':'Company_Type'}, ylabel='company_type'>

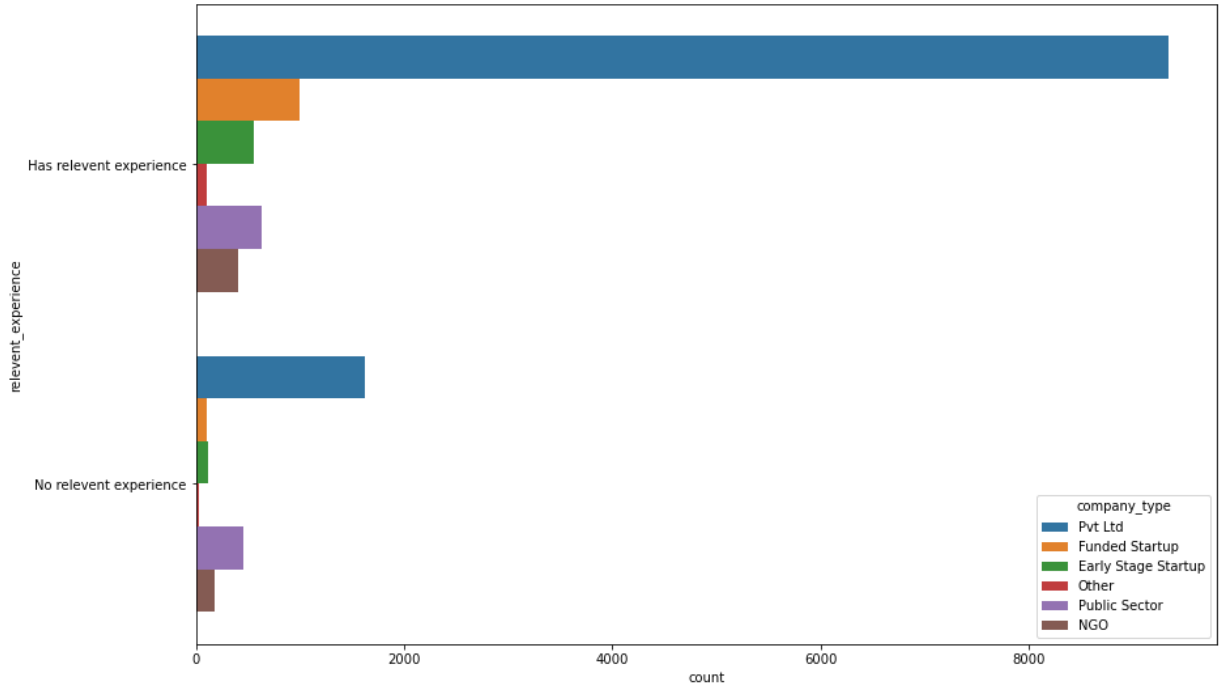



```
Maximum Peoples who are working in private companies are looking for a new job
```

Q10. Analysis of Relevant experience with company type

In [246]:
```
plt.figure(figsize=(14,9))
sns.countplot(y="relevent_experience",hue ='company_type',data=df)
```

Out[246]: <AxesSubplot:xlabel='count', ylabel='relevent_experience'>

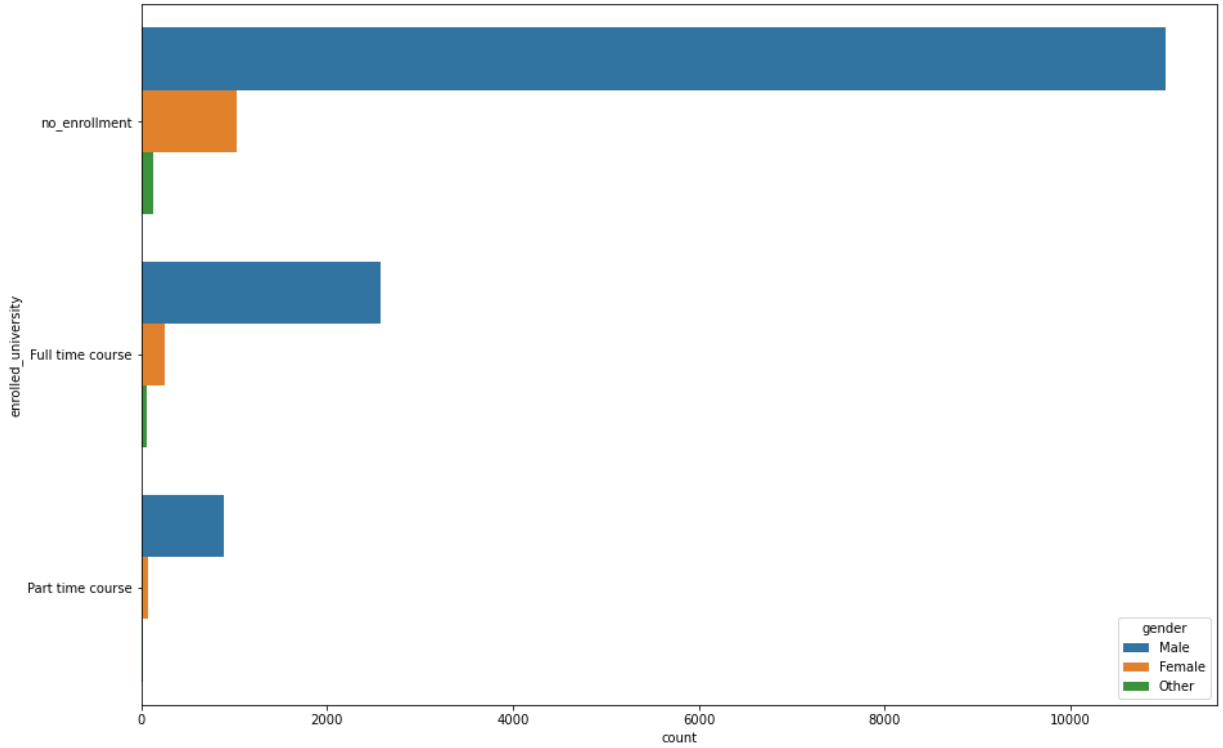


In [ ]:
```
People worked with private sector are having more relevant experience
```

Check whether Candidates are enrolled in a university or not

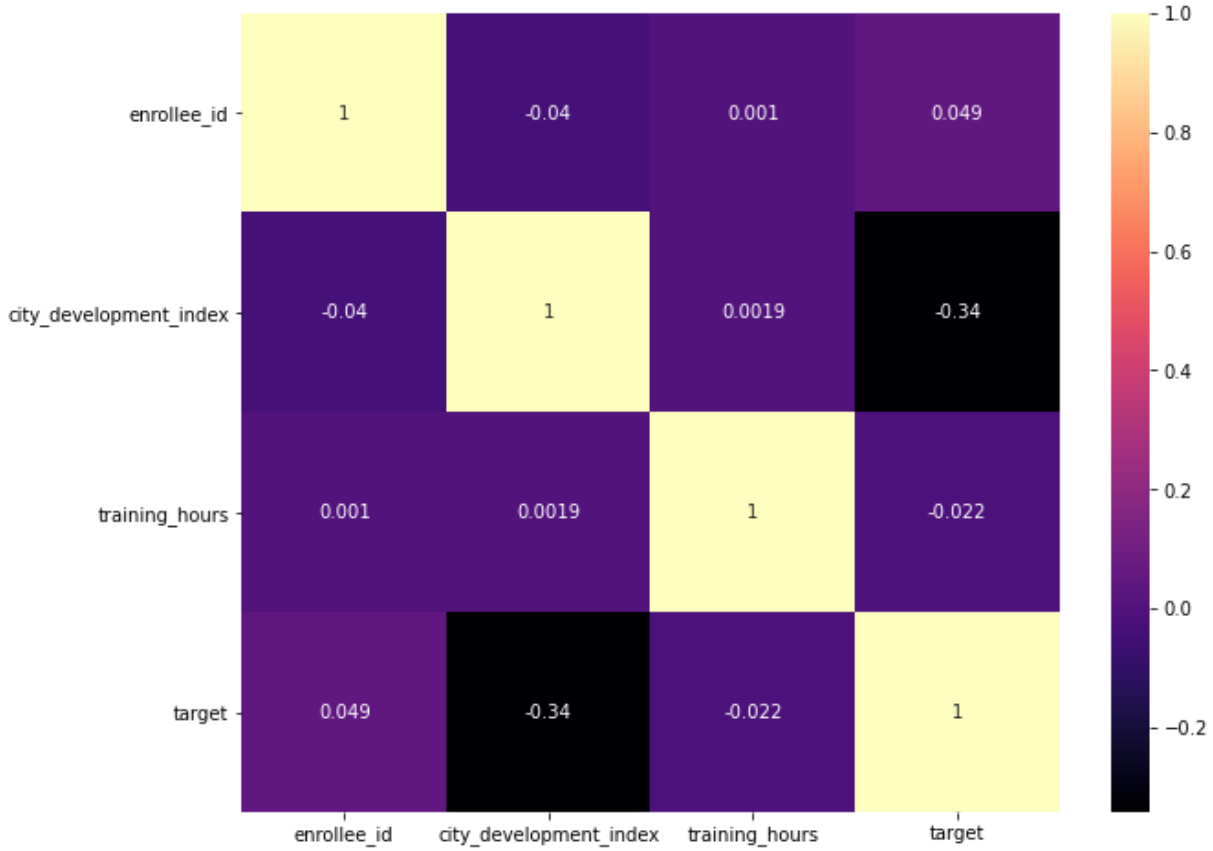
In [245]:
```
plt.figure(figsize=(15,10))
sns.countplot(y="enrolled_university",hue ='gender',data=df)
```

Out[245]: <AxesSubplot:xlabel='count', ylabel='enrolled_university'>



In [252]:
```
plt.figure(figsize=(10,8))
sns.heatmap(train.corr(),annot=True,cmap='magma')
```

Out[252]: <AxesSubplot:>



The correlation map

In [253]:
```
matrix = np.triu(train.corr())
plt.figure(figsize=(13, 10))
sns.heatmap(train.corr(), annot = True, cmap = 'YlOrBr', fmt=".2f", mask = matrix
            vmin = -1, vmax = 1, linewidths = 0.1, linecolor = 'white', cbar = Fa
plt.show()
```