Lab1

Exploratory Data Analysis (Bank Marketing)

2148059

Importing the libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
```

importing the dataset

In [2]:
```
bank=pd.read_csv("/Users/persie/Downloads/bank-12.csv")
bank.head(10)
```

Out[2]:

	age	job	marital	education	default	balance	housing	loan	contact	day	m
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	
5	35	management	single	tertiary	no	747	no	no	cellular	23	
6	36	self-employed	married	tertiary	no	307	yes	no	cellular	14	
7	39	technician	married	secondary	no	147	yes	no	cellular	6	
8	41	entrepreneur	married	tertiary	no	221	yes	no	unknown	14	
9	43	services	married	primary	no	-88	yes	yes	cellular	17	

checking null values

In [3]:
```
bank.isnull().sum()
```

Out[3]:
```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

There are no null values in the dataset

In [4]:
```
bank.describe()
```

Out[4]:

	age	balance	day	duration	campaign	pdays	
count	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	452
mean	41.170095	1422.657819	15.915284	263.961292	2.793630	39.766645	
std	10.576211	3009.638142	8.247667	259.856633	3.109807	100.121124	
min	19.000000	-3313.000000	1.000000	4.000000	1.000000	-1.000000	
25%	33.000000	69.000000	9.000000	104.000000	1.000000	-1.000000	
50%	39.000000	444.000000	16.000000	185.000000	2.000000	-1.000000	
75%	49.000000	1480.000000	21.000000	329.000000	3.000000	-1.000000	
max	87.000000	71188.000000	31.000000	3025.000000	50.000000	871.000000	2

In [5]:
```
bank.columns
```

Out[5]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'balance',
'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign',
'pdays',
       'previous', 'poutcome', 'y'],
      dtype='object')
```

In [6]:
```
bank.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        4521 non-null   int64 
 1   job        4521 non-null   object
 2   marital    4521 non-null   object
 3   education  4521 non-null   object
 4   default    4521 non-null   object
 5   balance    4521 non-null   int64 
 6   housing    4521 non-null   object
 7   loan       4521 non-null   object
 8   contact    4521 non-null   object
 9   day        4521 non-null   int64 
 10  month      4521 non-null   object
 11  duration   4521 non-null   int64 
 12  campaign   4521 non-null   int64 
 13  pdays      4521 non-null   int64 
 14  previous   4521 non-null   int64 
 15  poutcome   4521 non-null   object
 16  y          4521 non-null   object
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
```

In [7]:
```
classes=bank["y"]
yes=0
no=0
for cn,i in enumerate(classes):
    if i=="no":
        no+= 1
    if i=="yes":
        yes+=1
print("no of customers subscribed to the new term deposit:",yes,"\n
```

```
no of customers subscribed to the new term deposit: 521
no of customers did not subscribe: 4000
```

removing unwanted columns

In [8]:
```
rem=["contact","day"]
bank=bank.drop(rem,axis=1)
bank.columns
```

Out[8]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'balance',
'housing',
       'loan', 'month', 'duration', 'campaign', 'pdays', 'previous
',
       'poutcome', 'y'],
      dtype='object')
```

In [9]:
```
bank.head(3)
```

Out[9]:

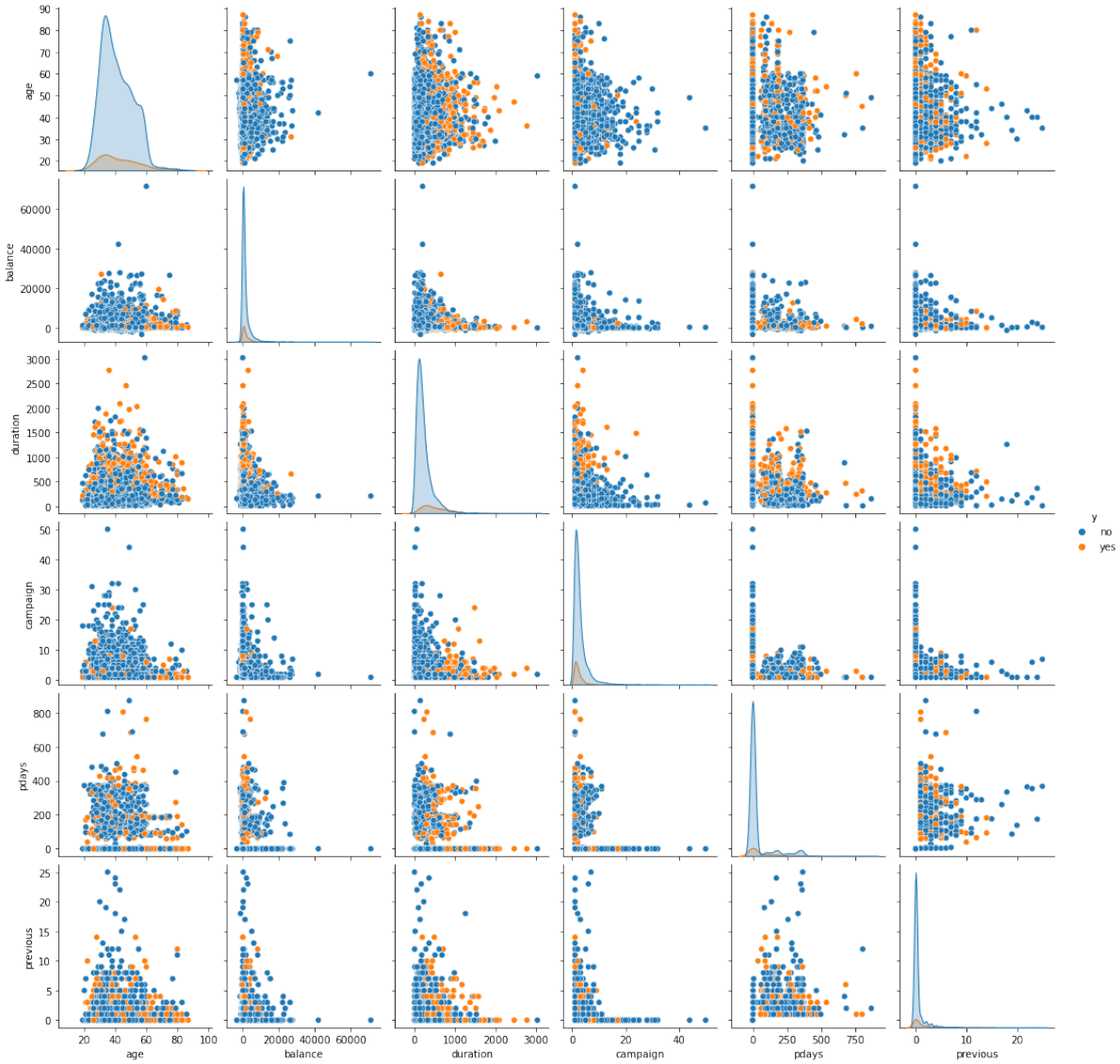
	age	job	marital	education	default	balance	housing	loan	month	duration
0	30	unemployed	married	primary	no	1787	no	no	oct	79
1	33	services	married	secondary	no	4789	yes	yes	may	220
2	35	management	single	tertiary	no	1350	yes	no	apr	185

plot the graph

In [10]:
```
plt.figure(figsize=(10,10))
sn.pairplot(bank, hue="y")
```

Out[10]:
```
<seaborn.axisgrid.PairGrid at 0x7f9280063310>

<Figure size 720x720 with 0 Axes>
```



In [71]:
```
b=bank.groupby(["poutcome","y"])["y"].count()
b=pd.DataFrame(b)
b
```

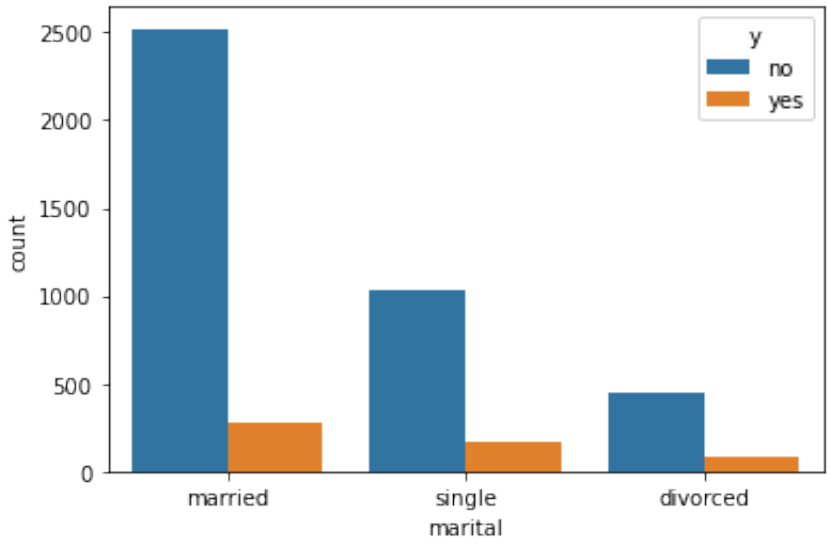
Out[71]:

		y
poutcome	**y**	
failure	**no**	427
	yes	63
other	**no**	159
	yes	38
success	**no**	46
	yes	83
unknown	**no**	3368
	yes	337

It shows that the customers who does not subscribe to the previous plan is still not convinced to the new plan. it also clearly shows that the customers who subscribed to the previous plan is hesitant to the new plan

In [12]:
```
sn.countplot(x=bank["marital"], hue=bank["y"])
```

Out[12]: <AxesSubplot:xlabel='marital', ylabel='count'>



The above graph shows that the overall subscribed rate is very low.

In [93]:
```
b=bank.groupby(["marital","y"]).size()
bk=pd.DataFrame(b)
bk
```

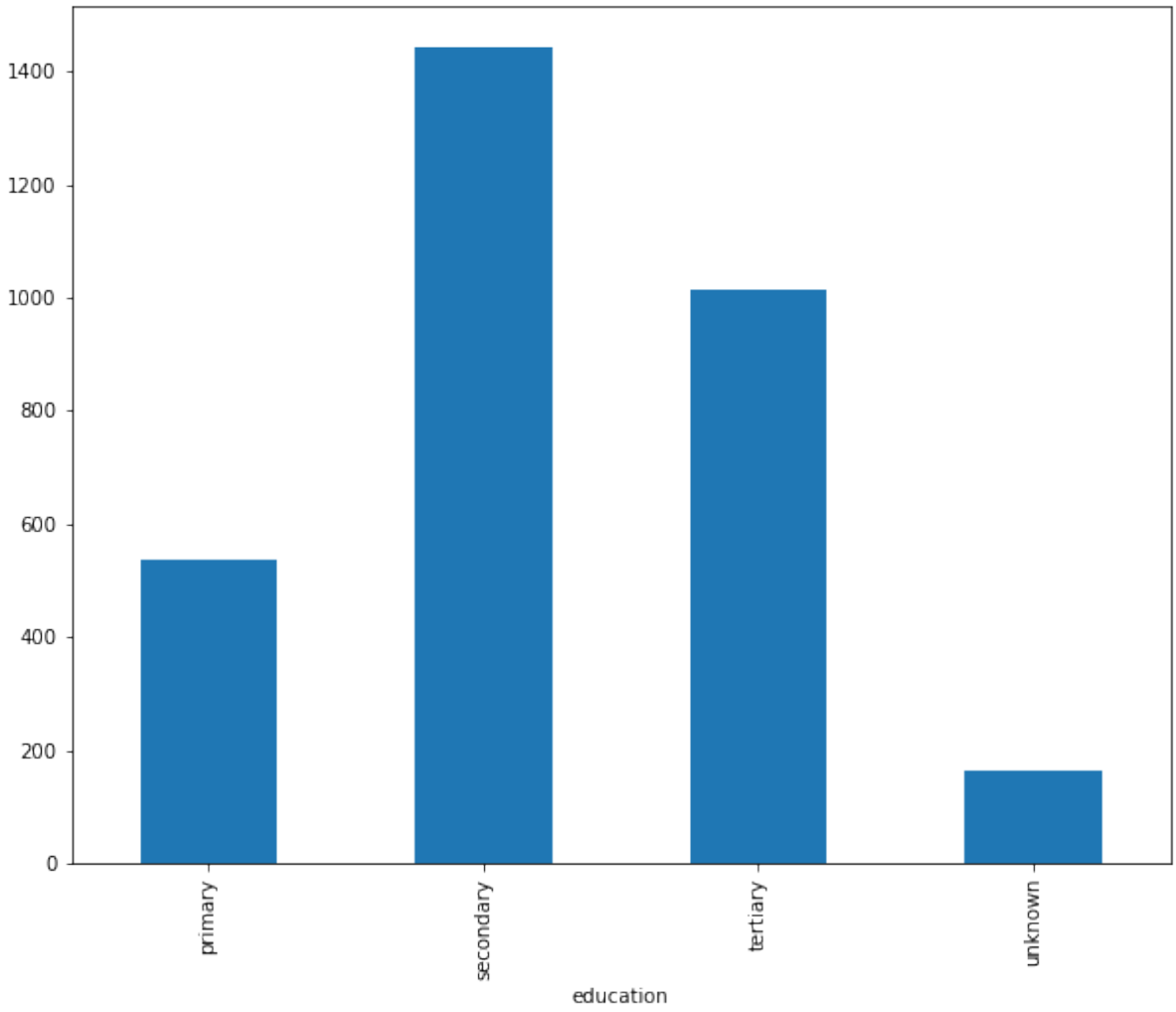
Out[93]:

		0
marital	**y**	
divorced	**no**	451
	yes	77
married	**no**	2520
	yes	277
single	**no**	1029
	yes	167

The subscription ratio of divorsed people is higher than single and married people

In [35]:
```
bank.groupby(["education"])["balance"].nunique().plot(kind="bar",fi
```

Out[35]: <AxesSubplot:xlabel='education'>



The people who had only secondary education are having high balance(in euros)

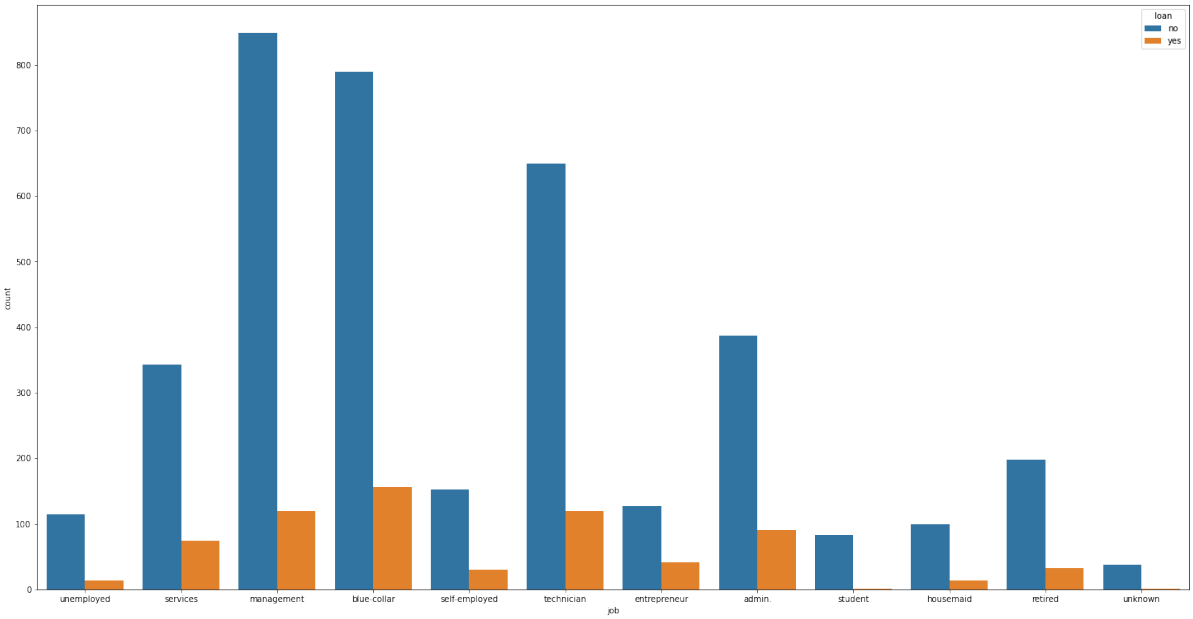
In [14]:
```
bank["education"].value_counts()
```

Out[14]:
```
secondary    2306
tertiary     1350
primary       678
unknown       187
Name: education, dtype: int64
```

the above graph is biased as people with secondary education is more presented in the dataset. there is a significant count difference between the education group.

In [102]:
```
plt.figure(figsize=(25,13))
sn.countplot(x="job", hue="loan",data=bank)
```

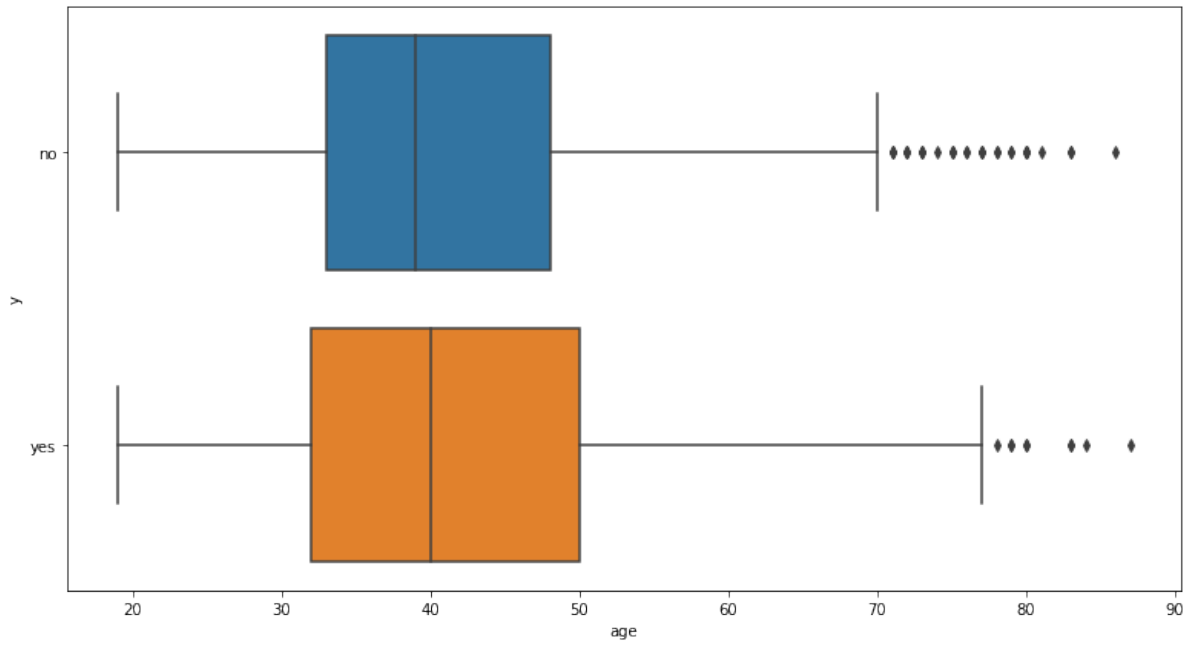
Out[102]: <AxesSubplot:xlabel='job', ylabel='count'>



This graph shows that high percentsage of customers in every sector are still without loan.

In [104]:
```
plt.figure(figsize=(13,7))
sn.boxplot(x=bank["age"],y="y",data=bank)
```

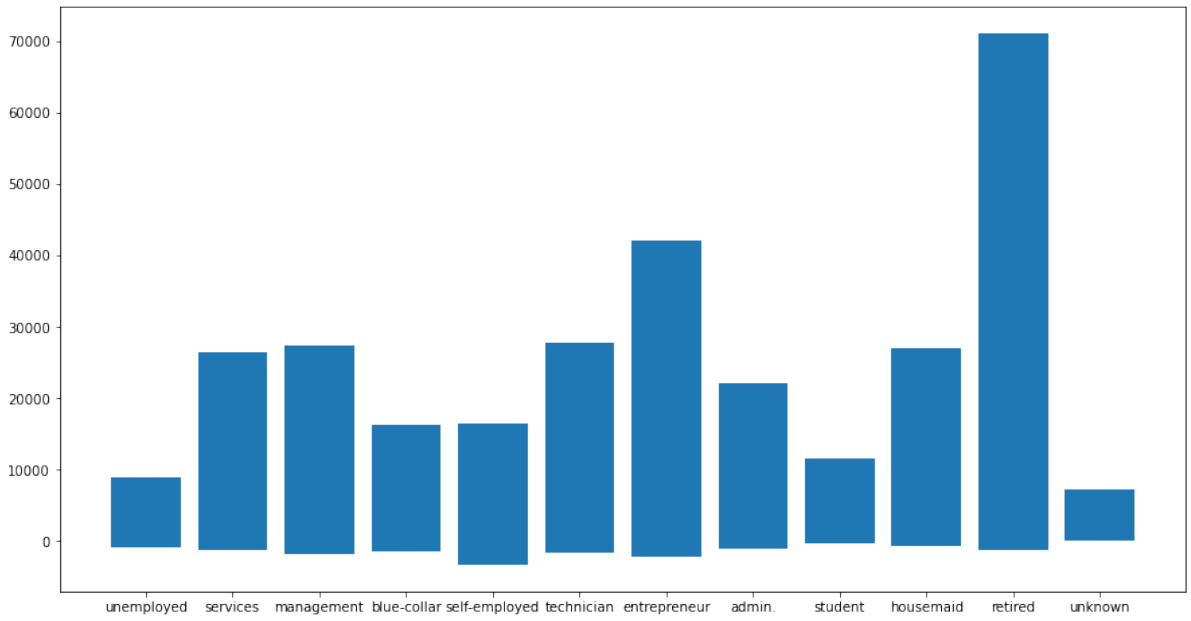
Out[104]: <AxesSubplot:xlabel='age', ylabel='y'>



It this interesting to observe that the range and variability of two outcomes are similar with respect to age.

In [17]:
```
plt.figure(figsize=(15,8))
plt.bar(x="job",height="balance",data=bank)
```

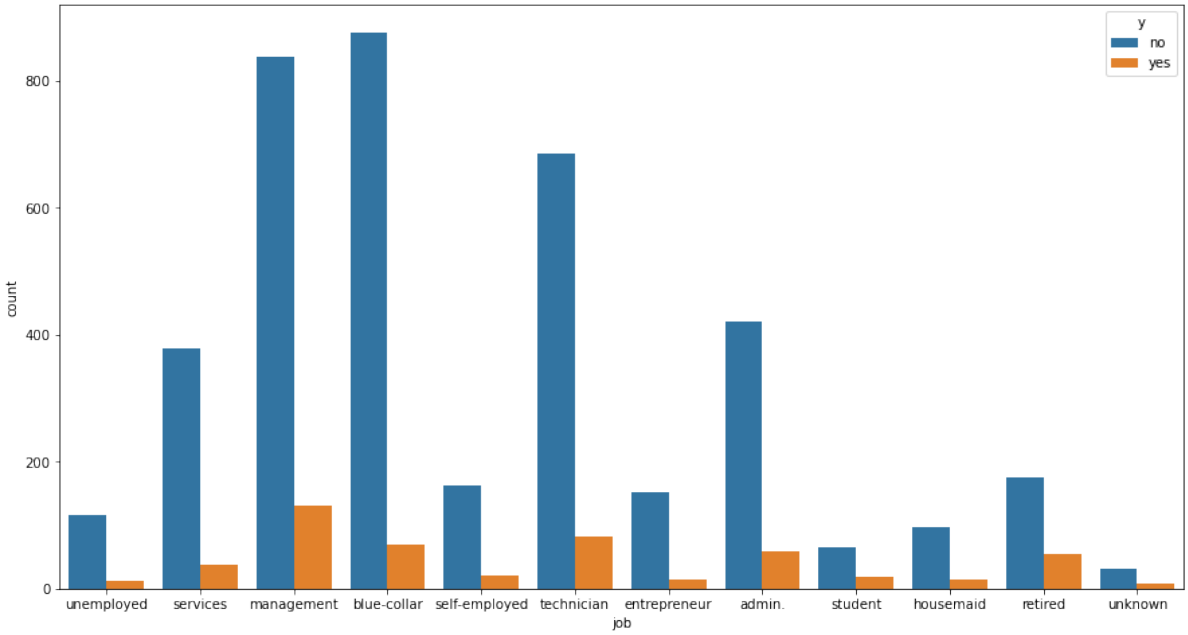
Out[17]:
```
<BarContainer object of 4521 artists>
```



The graphs shows that the retired clients and entrepreneurs have more balance

In [19]:
```
plt.figure(figsize=(15,8))
sn.countplot(x=bank["job"],hue="y",data=bank)
```

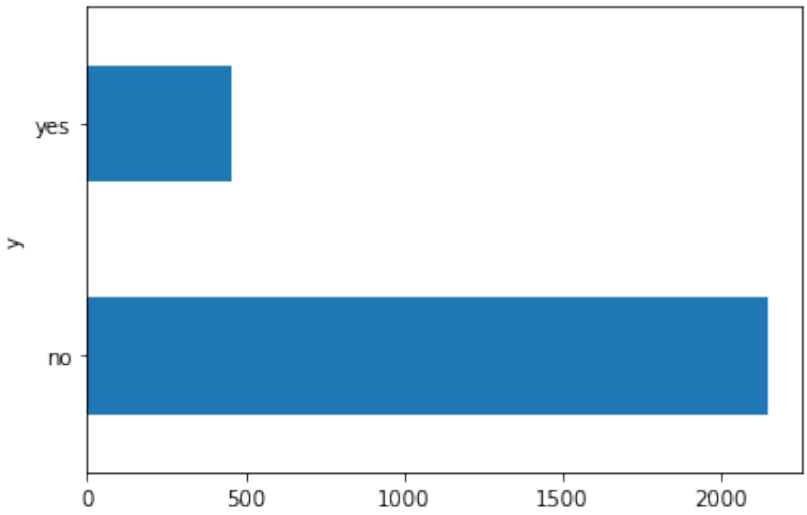
Out[19]: <AxesSubplot:xlabel='job', ylabel='count'>



Clients who are Blue collar workers, management professionals and technicians are most likely to respond 'no' to the new product.

In [105]:
```
bank.groupby("y")["balance"].nunique().plot(kind="barh")
```

Out[105]: <AxesSubplot:ylabel='y'>



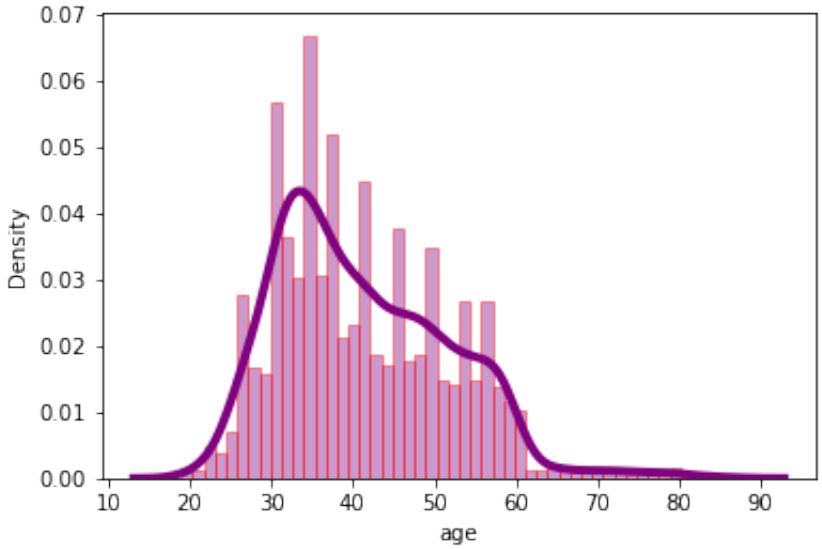
Clearly clients with more balance in account does not require the new term deposit.

histogram on numerical attribute 'age'

In [53]:
```
sn.distplot(bank["age"], hist=True,
            kde=True,bins=int(50),color='purple',
            hist_kws={"edgecolor":"red"},kde_kws={"linewidth":4})
```

```
/Users/persie/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[53]: <AxesSubplot:xlabel='age', ylabel='Density'>



The age data is right skewed

In [ ]: