In [1]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
df=pd.read_csv("/Users/persie/Downloads/train.csv")
df.head()
```

Out[2]:

	Id	age	job	marital	education	default	housing	loan	contact
0	25611	49	blue-collar	married	basic.9y	unknown	no	no	cellular
1	26010	37	entrepreneur	married	university.degree	no	no	no	telephone
2	40194	78	retired	married	basic.4y	no	no	no	cellular
3	297	36	admin.	married	university.degree	no	yes	no	telephone
4	36344	59	retired	divorced	university.degree	no	no	no	cellular

5 rows × 22 columns

In [3]:
```python
df.describe()
```

Out[3]:

	Id	age	duration	campaign	pdays	previou
count	32950.000000	32950.000000	32950.000000	32950.000000	32950.000000	32950.00000
mean	20618.796601	40.014112	258.127466	2.560607	962.052413	0.17471
std	11899.673392	10.403636	258.975917	2.752326	187.951096	0.49902
min	0.000000	17.000000	0.000000	1.000000	0.000000	0.00000
25%	10315.250000	32.000000	103.000000	1.000000	999.000000	0.00000
50%	20632.500000	38.000000	180.000000	2.000000	999.000000	0.00000
75%	30952.750000	47.000000	319.000000	3.000000	999.000000	0.00000
max	41187.000000	98.000000	4918.000000	56.000000	999.000000	7.00000

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Id                0
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
y                 0
dtype: int64
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32950 entries, 0 to 32949
Data columns (total 22 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Id              32950 non-null  int64  
 1   age             32950 non-null  int64  
 2   job             32950 non-null  object 
 3   marital         32950 non-null  object 
 4   education       32950 non-null  object 
 5   default         32950 non-null  object 
 6   housing         32950 non-null  object 
 7   loan            32950 non-null  object 
 8   contact         32950 non-null  object 
 9   month           32950 non-null  object 
 10  day_of_week     32950 non-null  object 
 11  duration        32950 non-null  int64  
 12  campaign        32950 non-null  int64  
 13  pdays           32950 non-null  int64  
 14  previous        32950 non-null  int64  
 15  poutcome        32950 non-null  object 
 16  emp.var.rate    32950 non-null  float64
 17  cons.price.idx  32950 non-null  float64
 18  cons.conf.idx   32950 non-null  float64
 19  euribor3m       32950 non-null  float64
 20  nr.employed     32950 non-null  float64
 21  y               32950 non-null  object 
dtypes: float64(5), int64(6), object(11)
memory usage: 5.5+ MB
```

In [6]:
```
df.drop(['Id'],axis=1,inplace=True)
```

In [7]:
```
df.head()
```

Out[7]:

	age	job	marital	education	default	housing	loan	contact	month
0	49	blue-collar	married	basic.9y	unknown	no	no	cellular	nov
1	37	entrepreneur	married	university.degree	no	no	no	telephone	nov
2	78	retired	married	basic.4y	no	no	no	cellular	jul
3	36	admin.	married	university.degree	no	yes	no	telephone	may
4	59	retired	divorced	university.degree	no	no	no	cellular	jun

5 rows × 21 columns

In [8]:
```
df.columns
```

Out[8]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'housing',
'loan',
       'contact', 'month', 'day_of_week', 'duration', 'campaign',
'pdays',
       'previous', 'poutcome', 'emp.var.rate', 'cons.price.idx',
       'cons.conf.idx', 'euribor3m', 'nr.employed', 'y'],
      dtype='object')
```

In [9]:
```
numeric=df.select_dtypes(include=np.number).columns.tolist()
```

In [10]:
```
category=df.select_dtypes(exclude=np.number).columns.tolist()
```

In [11]:
```
numeric
```

Out[11]:
```
['age',
 'duration',
 'campaign',
 'pdays',
 'previous',
 'emp.var.rate',
 'cons.price.idx',
 'cons.conf.idx',
 'euribor3m',
 'nr.employed']
```

In [12]:
```
category
```

Out[12]:
```
['job',
 'marital',
 'education',
 'default',
 'housing',
 'loan',
 'contact',
 'month',
 'day_of_week',
 'poutcome',
 'y']
```

In [13]:
```
num=df[numeric]
```

In [14]:
```
cat=df[category]
```

In [15]:
```
num.head()
```

Out[15]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	e
0	49	227	4	999	0	-0.1	93.200	-42.0	
1	37	202	2	999	1	-0.1	93.200	-42.0	
2	78	1148	1	999	0	-1.7	94.215	-40.3	
3	36	120	2	999	0	1.1	93.994	-36.4	
4	59	368	2	999	0	-2.9	92.963	-40.8	

handling missing data

In [16]:
```
df['age']=df['age'].fillna(df['age'].mean())
```

In [17]:
```
#mean with continuous data
df['duration']=df['duration'].fillna(df['duration'].median())
```

In [18]:
```
#with categorical data
df['loan']=df['loan'].fillna(df['loan'].mode())
```

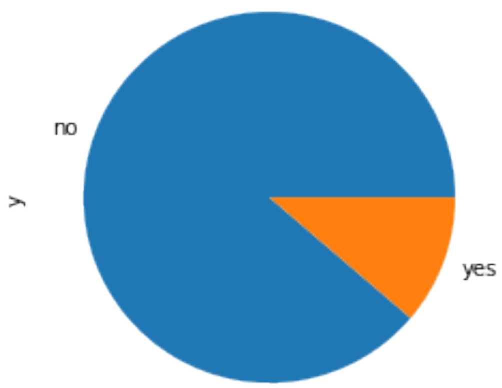
visualisation

In [19]:
```
df['y'].value_counts()
```

Out[19]:
```
no     29238
yes     3712
Name: y, dtype: int64
```

In [20]:
```
df['y'].value_counts().plot.pie()
```

Out[20]:
```
<AxesSubplot:ylabel='y'>
```

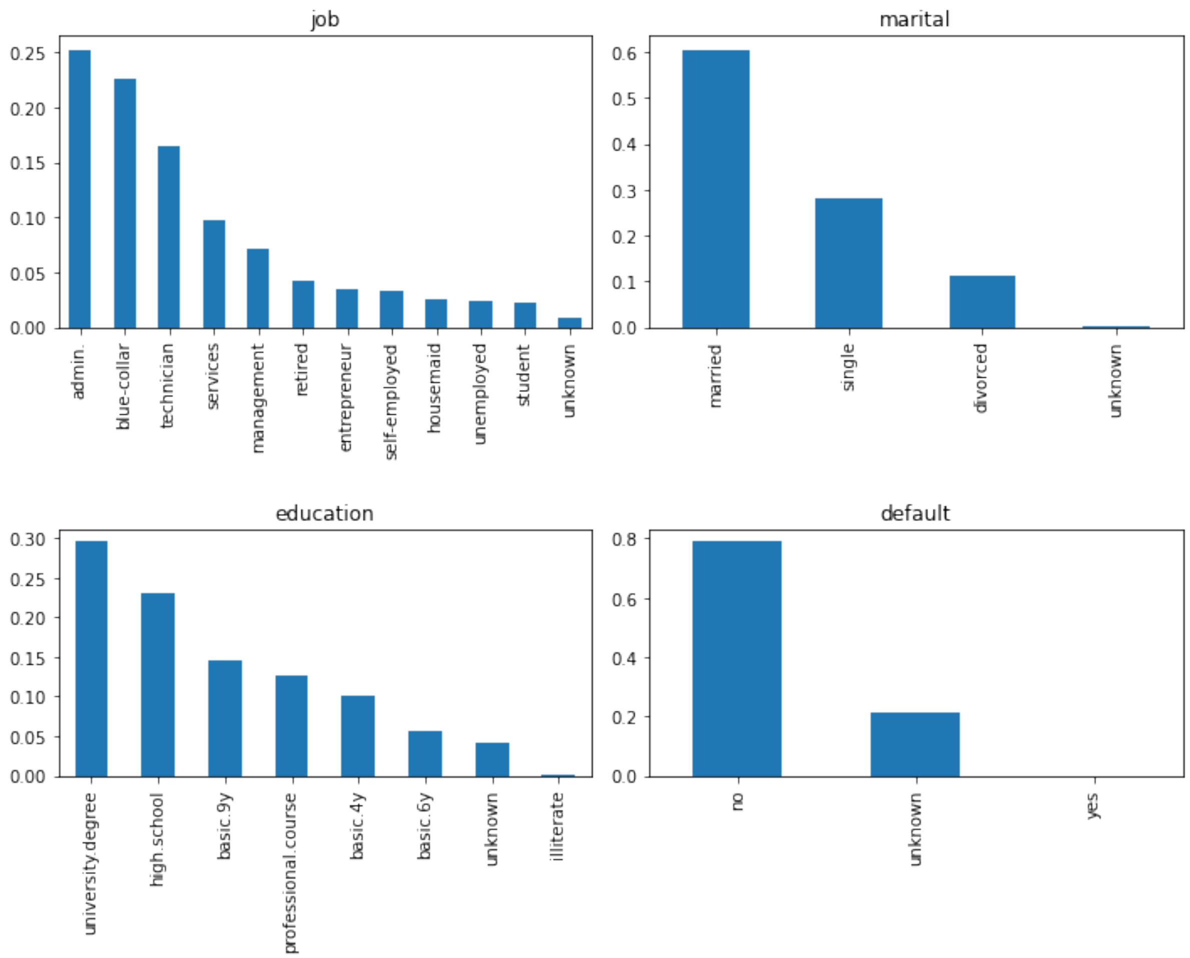


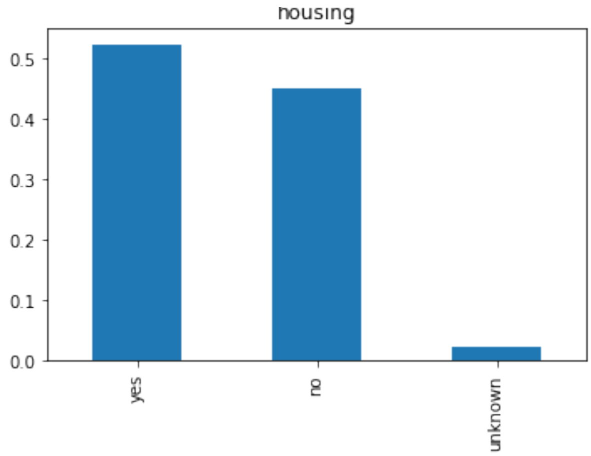
the data is imbalance

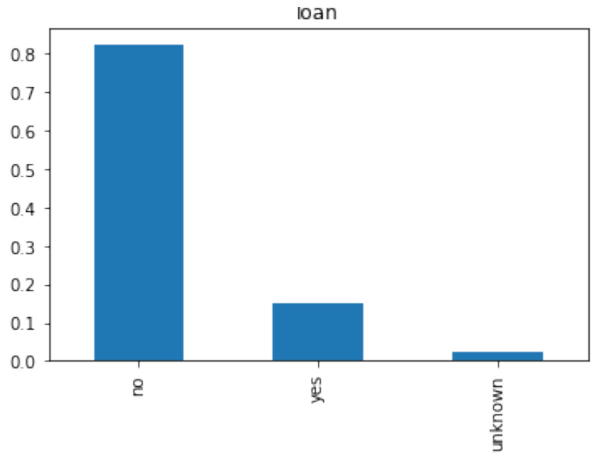
In [21]:
```
import matplotlib.pyplot as plt
```

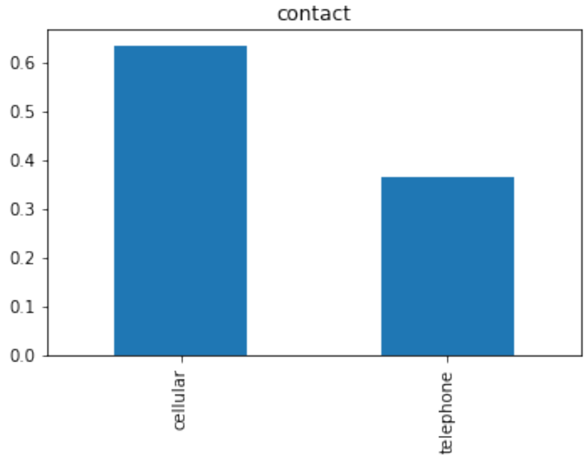
In [22]:
```
# Function to perform univariate analysis of categorical columns
def plot_categorical_columns(dataframe):
    categorical_columns = dataframe.select_dtypes(include=['object'

    for i in range(0,len(categorical_columns),2):
            if len(categorical_columns) > i+1:

                plt.figure(figsize=(10,4))
                plt.subplot(121)
                dataframe[categorical_columns[i]].value_counts(norm
                plt.title(categorical_columns[i])
                plt.subplot(122)
                dataframe[categorical_columns[i+1]].value_counts(no
                plt.title(categorical_columns[i+1])
                plt.tight_layout()
                plt.show()

            else:
                dataframe[categorical_columns[i]].value_counts(norm
                plt.title(categorical_columns[i])
```

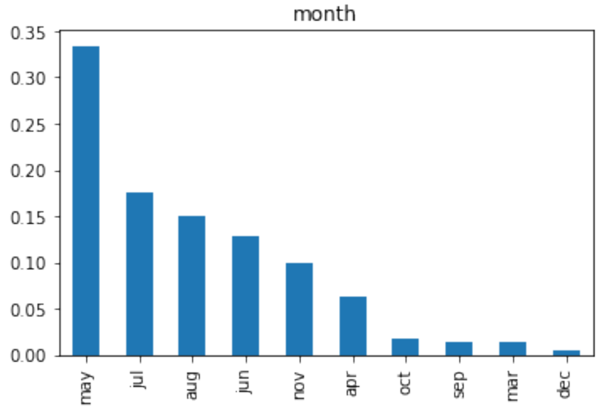
In [23]:
```
plot_categorical_columns(cat)
```

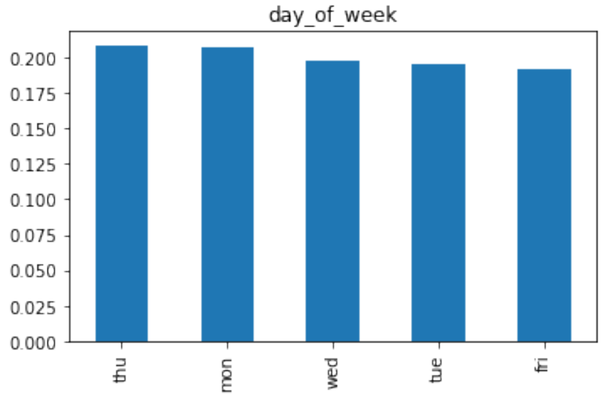


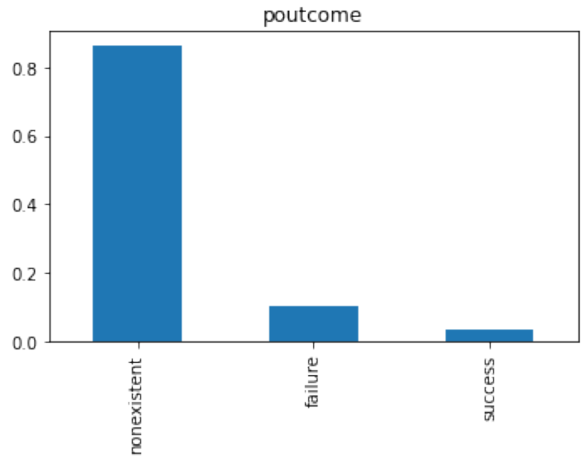

housing
0.0
0.1
0.2
0.3
0.4
0.5
yes
no
unknown

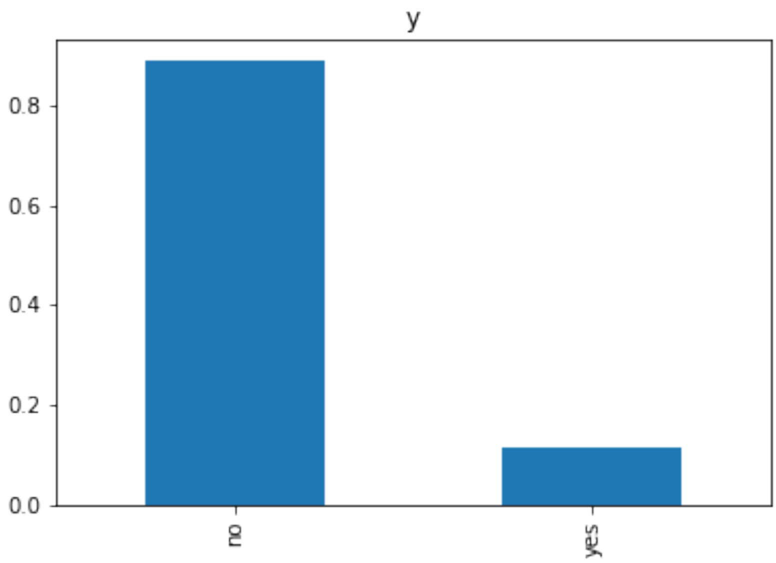

loan
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
no
yes
unknown


contact
0.0
0.1
0.2
0.3
0.4
0.5
0.6
cellular
telephone


month
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
may
jul
aug
jun
nov
apr
oct
sep
mar
dec


day_of_week
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
thu
mon
wed
tue
fri


poutcome
0.0
0.2
0.4
0.6
0.8
nonexistent
failure
success

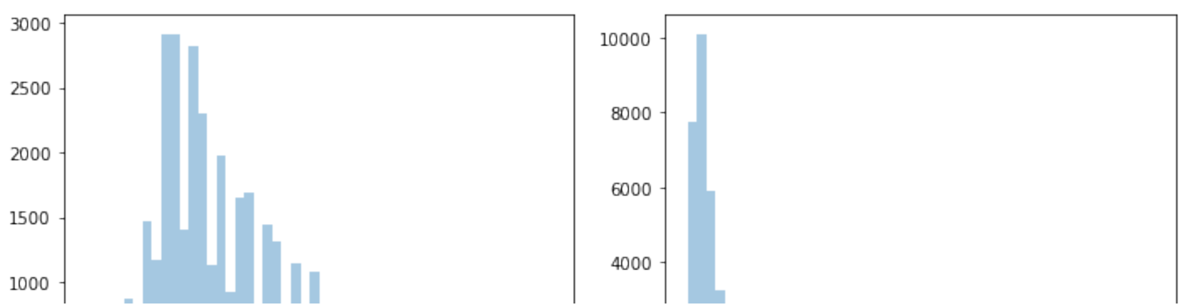

y
0.0
0.2
0.4
0.6
0.8
no
yes

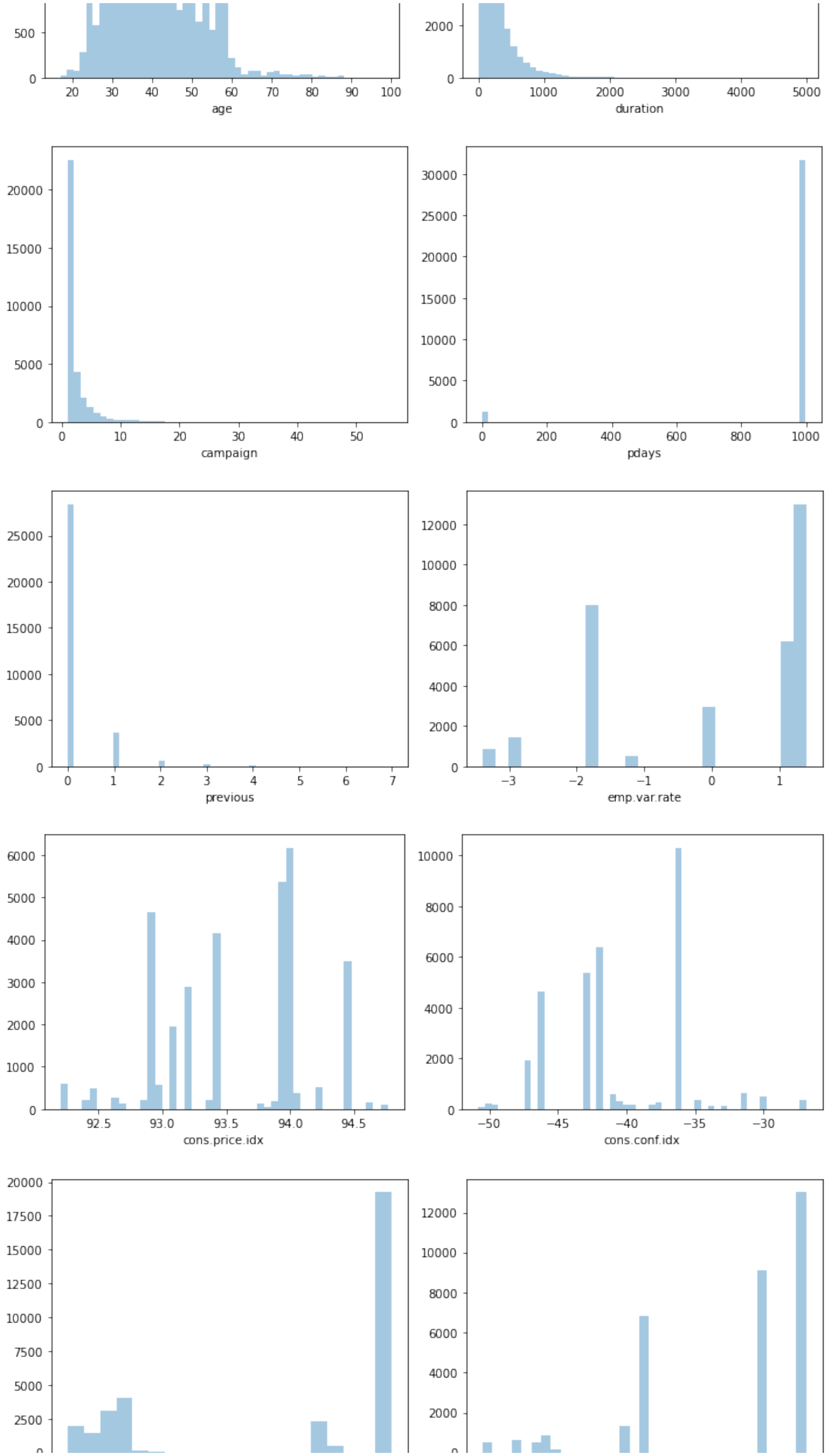
In [24]:
```python
import seaborn as sns
```

In [25]:
```python
# Function to plot histograms
def plot_continuous_columns(dataframe):
    numeric_columns = dataframe.select_dtypes(include=['number']).c
    dataframe = dataframe[numeric_columns]

    for i in range(0,len(numeric_columns),2):
        if len(numeric_columns) > i+1:
            plt.figure(figsize=(10,4))
            plt.subplot(121)
            sns.distplot(dataframe[numeric_columns[i]], kde=False)
            plt.subplot(122)
            sns.distplot(dataframe[numeric_columns[i+1]], kde=False
            plt.tight_layout()
            plt.show()

        else:
            sns.distplot(dataframe[numeric_columns[i]], kde=False)

# Function to plot boxplots
def plot_box_plots(dataframe):
    numeric_columns = dataframe.select_dtypes(include=['number']).c
    dataframe = dataframe[numeric_columns]

    for i in range(0,len(numeric_columns),2):
        if len(numeric_columns) > i+1:
            plt.figure(figsize=(10,4))
            plt.subplot(121)
            sns.boxplot(dataframe[numeric_columns[i]])
            plt.subplot(122)
            sns.boxplot(dataframe[numeric_columns[i+1]])
            plt.tight_layout()
            plt.show()

        else:
            sns.boxplot(dataframe[numeric_columns[i]])
```

In [26]:
```python
plot_continuous_columns(num)
```

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

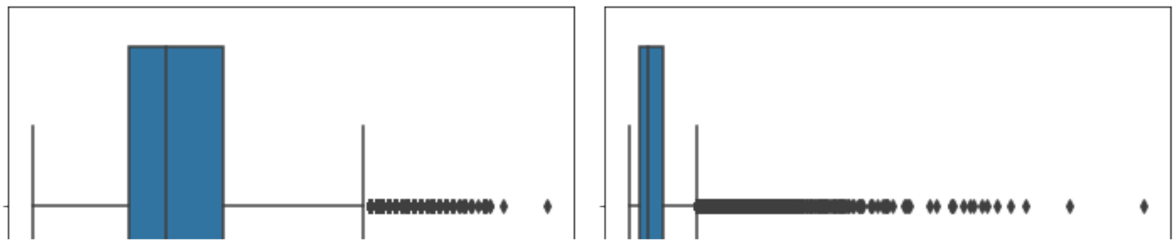




age
duration
campaign
pdays
previous
emp.var.rate
cons.price.idx
cons.conf.idx

In [27]:
```
plot_box_plots(num)
```

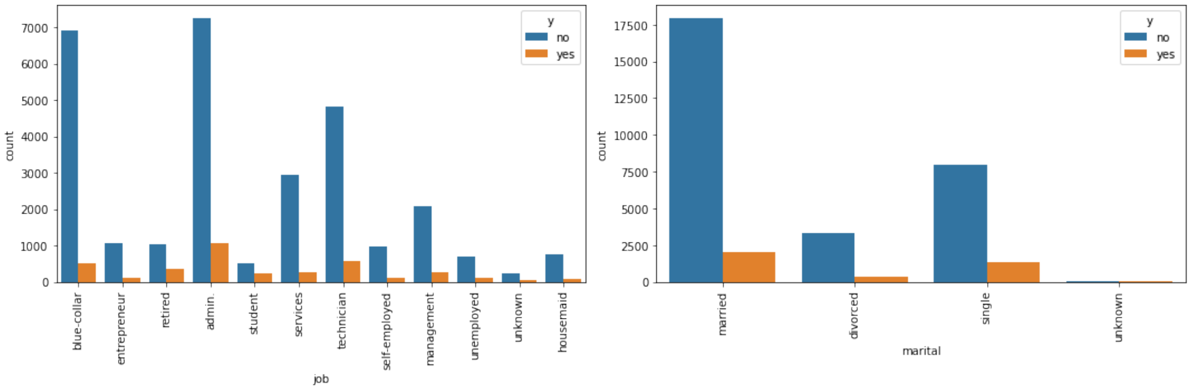
```
/Users/persie/opt/anaconda3/lib/python3.8/site-packages/seaborn/_d
ecorators.py:36: FutureWarning: Pass the following variable as a k
eyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
/Users/persie/opt/anaconda3/lib/python3.8/site-packages/seaborn/_d
ecorators.py:36: FutureWarning: Pass the following variable as a k
eyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [28]:
```
def bivariate_analysis_categorical(dataframe,target):
    categorical_columns = dataframe.select_dtypes(exclude=np.number
    for i in range(0,len(categorical_columns),2):
        if len(categorical_columns) > i+1:
            plt.figure(figsize=(15,5))
            plt.subplot(121)
            sns.countplot(x=dataframe[categorical_columns[i]],hue=t
            plt.xticks(rotation=90)
            plt.subplot(122)
            sns.countplot(dataframe[categorical_columns[i+1]],hue=t
            plt.xticks(rotation=90)
            plt.tight_layout()
            plt.show()
```

In [29]:
```
bivariate_analysis_categorical(cat,df['y'])
```

```
/Users/persie/opt/anaconda3/lib/python3.8/site-packages/seaborn/_d
ecorators.py:36: FutureWarning: Pass the following variable as a k
eyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```




```
/Users/persie/opt/anaconda3/lib/python3.8/site-packages/seaborn/_d
```

In [30]:
```
df.drop(['day_of_week'],axis=1,inplace=True)
```

treat outliers

In [31]:
```
from scipy.stats.mstats import winsorize
# Function to treat outliers
def treat_outliers(dataframe):
    cols = list(dataframe)
    for col in cols:
        if col in dataframe.select_dtypes(include=np.number).column
            dataframe[col] = winsorize(dataframe[col], limits=[0.05

    return dataframe
```

In [32]:
```
df['y']=df['y'].map({'no':0,'yes':1})
```

In [33]:
```
x=df.drop(['y'],axis=1)
y=df['y']
```

In [34]:
```
df=treat_outliers(x)
```

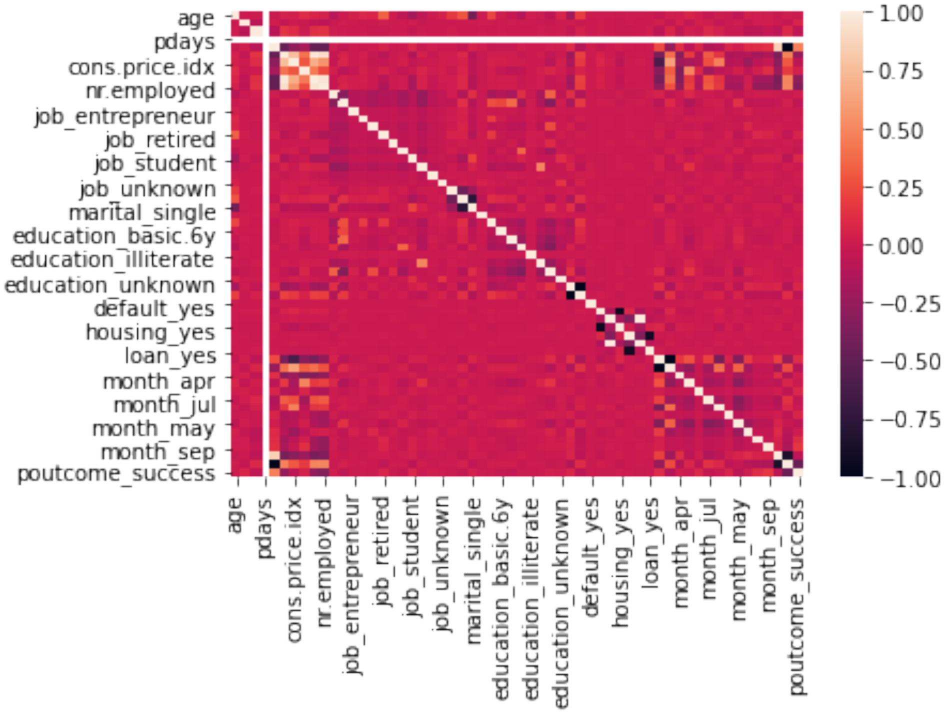
categorical into numerical

In [35]:
```
#label encoding
df=pd.get_dummies(df)
df.shape
```

Out[35]: (32950, 58)

In [36]:
```python
#pca,linear assuption : no multi correality
# from heatmap
sns.heatmap(df.corr())
```

Out[36]: <AxesSubplot:>



In [37]:
```python
df.corr()
```

Out[37]:

	age	duration	campaign	pdays	previous	emp.var.ra
age	1.000000	-0.000494	0.007063	NaN	-0.013049	0.0545
duration	-0.000494	1.000000	-0.080557	NaN	0.028897	-0.0476
campaign	0.007063	-0.080557	1.000000	NaN	-0.090701	0.1417
pdays	NaN	NaN	NaN	NaN	NaN	N
previous	-0.013049	0.028897	-0.090701	NaN	1.000000	-0.4754
emp.var.rate	0.054567	-0.047689	0.141779	NaN	-0.475454	1.0000
cons.price.idx	0.034671	0.002902	0.105762	NaN	-0.303604	0.7641
cons.conf.idx	0.109320	-0.011168	0.003443	NaN	-0.175793	0.3991
euribor3m	0.064946	-0.054867	0.122487	NaN	-0.490059	0.9755
nr.employed	0.045264	-0.071522	0.136186	NaN	-0.494280	0.9248
job_admin.	-0.097025	-0.012889	0.012549	NaN	0.013382	-0.0253
job_blue-collar	-0.002416	0.017245	0.002405	NaN	-0.045912	0.0579
job_entrepreneur	0.038408	0.005867	-0.004445	NaN	-0.006510	0.0066
job_housemaid	0.083648	-0.005830	0.001541	NaN	-0.010889	0.0382
job_management	0.075019	-0.003460	-0.013105	NaN	0.007649	-0.0180
job_retired	0.329492	0.016679	-0.012714	NaN	0.053050	-0.0981

job_self-employed	0.003900	0.002277	0.007785	NaN	-0.010414	-0.0004
job_services	-0.060988	0.003520	0.006882	NaN	-0.004994	0.0203
job_student	-0.191838	0.019115	-0.024938	NaN	0.084986	-0.1396
job_technician	-0.058022	-0.018815	0.003180	NaN	-0.018815	0.0537
job_unemployed	0.001843	-0.005689	-0.005099	NaN	0.014438	-0.0214
job_unknown	0.047288	-0.006619	-0.002047	NaN	-0.005390	0.0160
marital_divorced	0.165293	-0.003978	0.009458	NaN	-0.000753	0.0188
marital_married	0.287181	-0.003782	-0.005444	NaN	-0.044630	0.0835
marital_single	-0.428142	0.005610	-0.000937	NaN	0.047674	-0.1029
marital_unknown	0.001027	0.013051	0.002351	NaN	0.014010	-0.0113
education_basic.4y	0.226281	0.015128	-0.000327	NaN	-0.023442	0.0300
education_basic.6y	0.013173	0.002953	0.007771	NaN	-0.017777	0.0250
education_basic.9y	-0.029002	0.008312	-0.007204	NaN	-0.019702	0.0210
education_high.school	-0.105649	0.006022	-0.000751	NaN	0.021707	-0.0172
education_illiterate	0.015676	0.001104	-0.001043	NaN	-0.004805	-0.0021
education_professional.course	0.007177	-0.012272	0.008805	NaN	-0.006300	0.0213
education_university.degree	-0.069932	-0.016584	-0.003889	NaN	0.018402	-0.0473
education_unknown	0.063258	0.004562	0.000145	NaN	0.013891	-0.0024
default_no	-0.194819	0.014171	-0.040983	NaN	0.108960	-0.2071
default_unknown	0.194783	-0.013998	0.041129	NaN	-0.109104	0.2070
default_yes	0.002867	-0.007481	-0.005995	NaN	0.005422	0.0049
housing_no	0.003995	0.011294	0.011987	NaN	-0.029610	0.0594
housing_unknown	0.003025	-0.007001	-0.002584	NaN	-0.002600	0.0060
housing_yes	-0.004911	-0.009103	-0.011151	NaN	0.030307	-0.0611
loan_no	0.001621	0.006262	-0.010949	NaN	0.000626	-0.0007
loan_unknown	0.003025	-0.007001	-0.002584	NaN	-0.002600	0.0060
loan_yes	-0.003012	-0.003653	0.012719	NaN	0.000447	-0.0017
contact_cellular	-0.024952	0.034666	-0.065177	NaN	0.243322	-0.3945
contact_telephone	0.024952	-0.034666	0.065177	NaN	-0.243322	0.3945
month_apr	0.009039	0.043842	-0.057812	NaN	0.117642	-0.3189
month_aug	0.067164	-0.054125	0.026837	NaN	-0.077649	0.1776
month_dec	0.028803	0.032535	-0.006807	NaN	0.066104	-0.1232
month_jul	-0.037645	0.025516	0.084361	NaN	-0.138454	0.3159
month_jun	0.000270	-0.031417	0.046742	NaN	-0.087658	0.1500
month_mar	-0.013487	-0.001818	-0.010253	NaN	0.065539	-0.1420
month_may	-0.056170	0.021410	-0.007896	NaN	0.018651	-0.1228

month_nov	0.035956	-0.024554	-0.083451	NaN	0.101299	-0.0999
month_oct	0.018739	0.018414	-0.063842	NaN	0.111377	-0.1920
month_sep	0.007944	0.020090	-0.038161	NaN	0.130961	-0.1576
poutcome_failure	-0.018343	-0.005620	-0.068021	NaN	0.853216	-0.3863
poutcome_nonexistent	0.013049	-0.028897	0.090701	NaN	-1.000000	0.4754
poutcome_success	0.006143	0.064824	-0.058182	NaN	0.466338	-0.2544

58 rows × 58 columns

PCA

In [38]:
```
df.shape
```
Out[38]:
```
(32950, 58)
```

In [39]:
```
df.columns
```
Out[39]:
```
Index(['age', 'duration', 'campaign', 'pdays', 'previous', 'emp.va
r.rate',
       'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employe
d',
       'job_admin.', 'job_blue-collar', 'job_entrepreneur', 'job_h
ousemaid',
       'job_management', 'job_retired', 'job_self-employed', 'job_
services',
       'job_student', 'job_technician', 'job_unemployed', 'job_unk
nown',
       'marital_divorced', 'marital_married', 'marital_single',
       'marital_unknown', 'education_basic.4y', 'education_basic.6
y',
       'education_basic.9y', 'education_high.school', 'education_i
lliterate',
       'education_professional.course', 'education_university.degr
ee',
       'education_unknown', 'default_no', 'default_unknown', 'defa
ult_yes',
       'housing_no', 'housing_unknown', 'housing_yes', 'loan_no',
       'loan_unknown', 'loan_yes', 'contact_cellular', 'contact_te
lephone',
       'month_apr', 'month_aug', 'month_dec', 'month_jul', 'month_
jun',
       'month_mar', 'month_may', 'month_nov', 'month_oct', 'month_
sep',
       'poutcome_failure', 'poutcome_nonexistent', 'poutcome_succe
ss'],
      dtype='object')
```

In [40]:
```
from sklearn.decomposition import PCA
```

In [41]:
```
pca=PCA(n_components=10)
```

In [42]:
```
pca.fit(df)
```
Out[42]:
```
PCA(n_components=10)
```
In [43]:
```
X=pca.transform(df)
```
In [44]:
```
X.shape
```
Out[44]:
```
(32950, 10)
```
In [45]:
```
pca.singular_values_
```
Out[45]:
```
array([28835.08560229, 12648.4455777 ,  1649.04292151,   702.89559
497,
         244.71471825,   160.09581335,   125.67873386,   118.48668
801,
         104.88992672,   101.33245981])
```
In [46]:
```
pca.explained_variance_ratio_
```
Out[46]:
```
array([8.35732656e-01, 1.60804869e-01, 2.73330944e-03, 4.96599783e
-04,
       6.01928717e-05, 2.57623118e-05, 1.58762700e-05, 1.41112018e
-05,
       1.10584038e-05, 1.03210065e-05])
```
LDA

In [47]:
```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysi
```
In [49]:
```
lda=LDA(n_components=1)
lda.fit_transform(df,y)
```
Out[49]:
```
array([[-0.36392791],
       [-0.83995188],
       [ 3.84527335],
       ...,
       [-0.75295795],
       [-0.760131  ],
       [ 1.27362264]])
```
In [58]:
```
pred=lda.predict(df)
```
In [61]:
```
from sklearn.metrics import confusion_matrix, classification_report
```
In [62]:
```
confusion_matrix(pred,y)
```
Out[62]:
```
array([[28265,  2140],
       [  973,  1572]])
```

In [63]:
```
print(classification_report(pred,y))
```

```
              precision    recall  f1-score   support

           0       0.97      0.93      0.95     30405
           1       0.42      0.62      0.50      2545

    accuracy                           0.91     32950
   macro avg       0.70      0.77      0.73     32950
weighted avg       0.92      0.91      0.91     32950
```

In [ ]: