```
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
from sklearn.tree import plot_tree
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
df = pd.read_csv('/content/sample_data/bank-additional-full.csv',delimiter=';')
df.rename(columns={'y':'deposit'},inplace=True)
```

```
df.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcor
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	...	1	999	0	nonexiste
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	...	1	999	0	nonexiste
2	37	services	married	high.school	no	yes	no	telephone	may	mon	...	1	999	0	nonexiste
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	...	1	999	0	nonexiste
4	56	services	married	high.school	no	no	yes	telephone	may	mon	...	1	999	0	nonexiste

5 rows × 21 columns

```
df.tail()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous
41183	73	retired	married	professional.course	no	yes	no	cellular	nov	fri	...	1	999	0
41184	46	blue-collar	married	professional.course	no	no	no	cellular	nov	fri	...	1	999	0
41185	56	retired	married	university.degree	no	yes	no	cellular	nov	fri	...	2	999	0
41186	44	technician	married	professional.course	no	no	no	cellular	nov	fri	...	1	999	0
41187	74	retired	married	professional.course	no	yes	no	cellular	nov	fri	...	3	999	1

```
df.shape
```

```
(41188, 21)
```

```
df.dtypes
```

	0
age	int64
job	object
marital	object
education	object
default	object
housing	object
loan	object
contact	object
month	object
day_of_week	object
duration	int64
campaign	int64
pdays	int64
previous	int64
poutcome	object
emp.var.rate	float64
cons.price.idx	float64
cons.conf.idx	float64
euribor3m	float64
nr.employed	float64
deposit	object

dtype: object

```
df.columns
```

```
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', 'deposit'],
      dtype='object')
```

```
df.dtypes.value_counts()
```

	count
object	11
int64	5
float64	5

dtype: int64

```
df.duplicated().sum()
```

```
12
```

```
df.isna().sum()
```

	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
deposit	0

dtype: int64

```
cat_cols = df.select_dtypes(include='object').columns
print(cat_cols)
num_cols = df.select_dtypes(include='number').columns
print(num_cols)
```

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

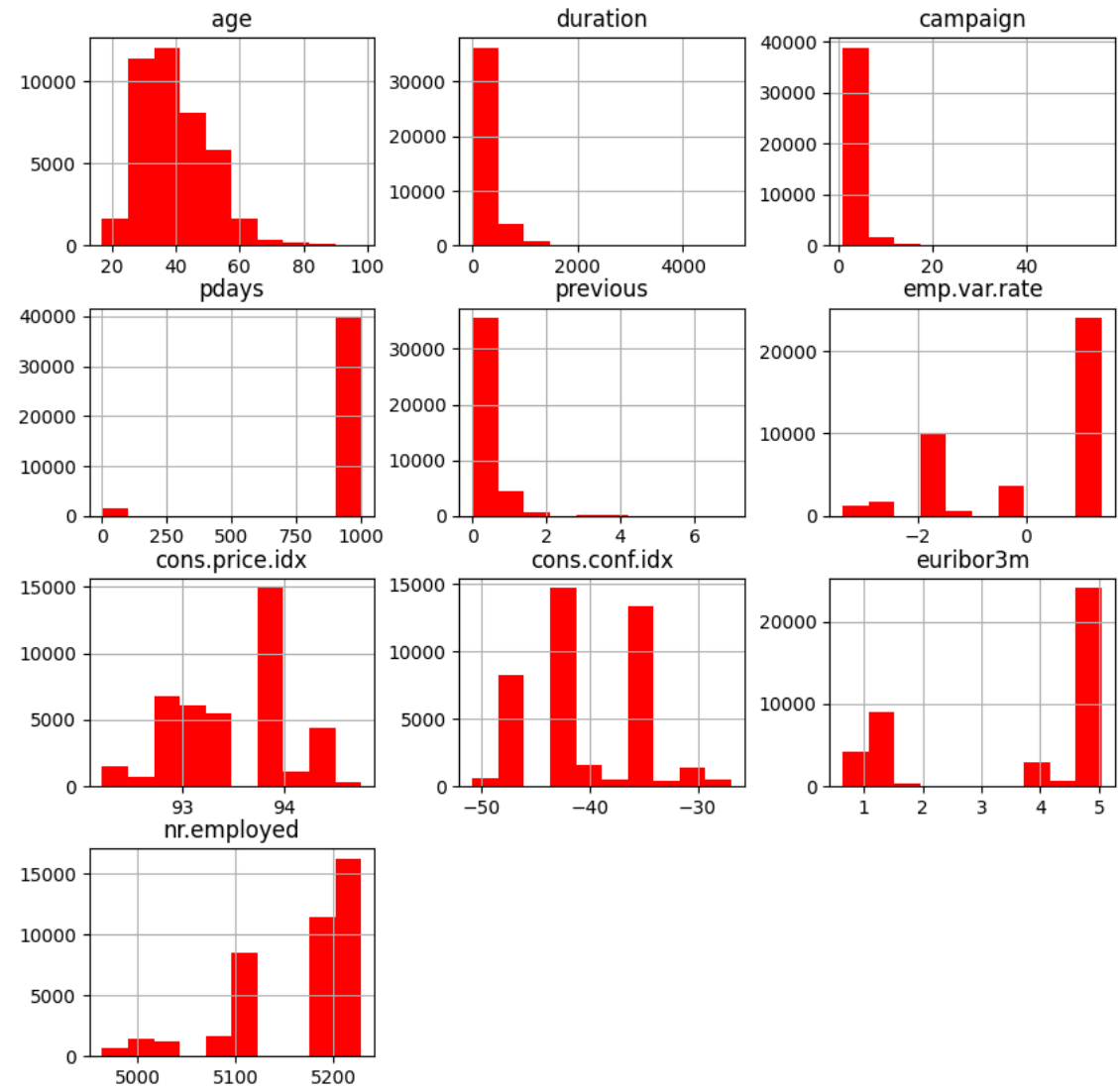
```
df.describe()
```

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000

```
df.describe(include='object')
```

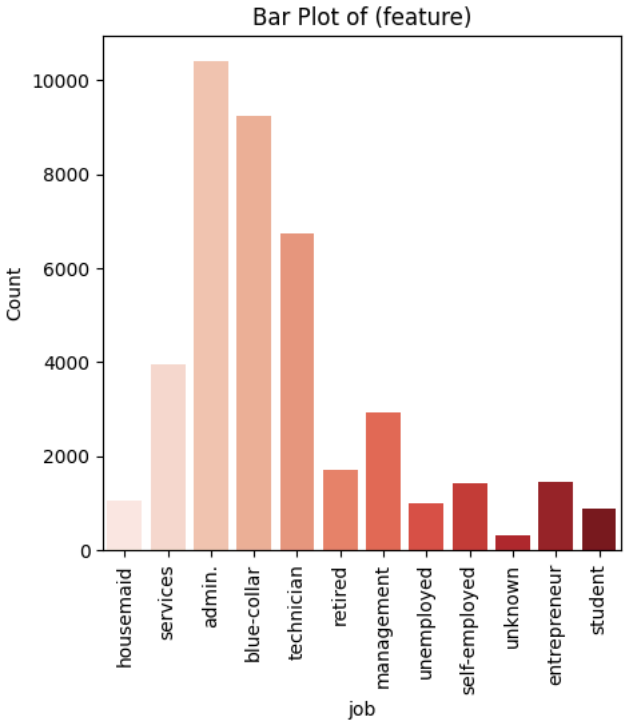
	job	marital	education	default	housing	loan	contact	month	day_of_week	poutcome	deposit
count	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188
unique	12	4	8	3	3	3	2	10	5	3	2
top	admin.	married	university.degree	no	yes	no	cellular	may	thu	nonexistent	no
freq	10422	24928	12168	32588	21576	33950	26144	13769	8623	35563	36548

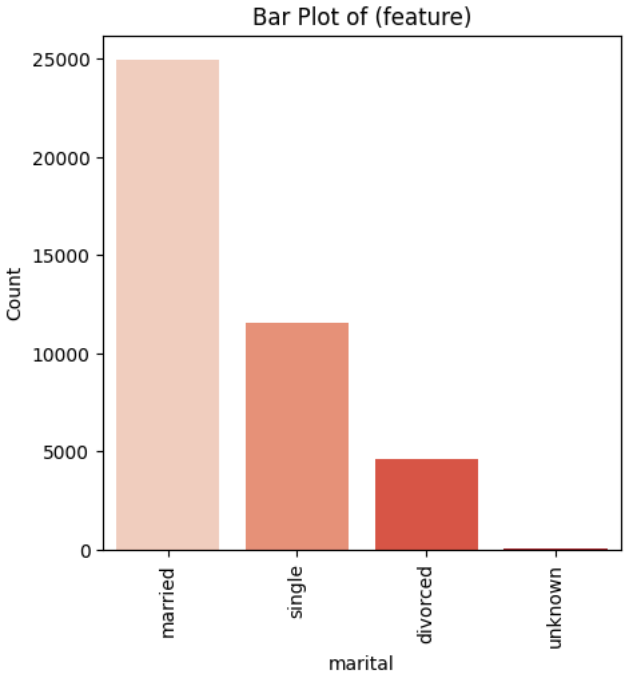
```
df.hist(figsize=(10,10), color='red')
plt.show()
```

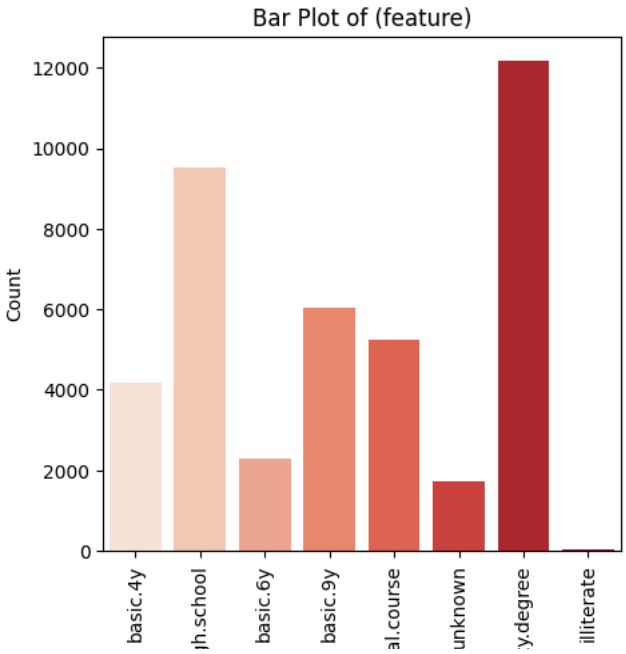



```
for feature in cat_cols:
    plt.figure(figsize=(5,5))
    sns.countplot(x=feature,data=df, palette='Reds')
    plt.title(f'Bar Plot of (feature)')
    plt.xlabel(f'{feature}')
    plt.ylabel('Count')
    plt.xticks(rotation=90)
    plt.show()
```

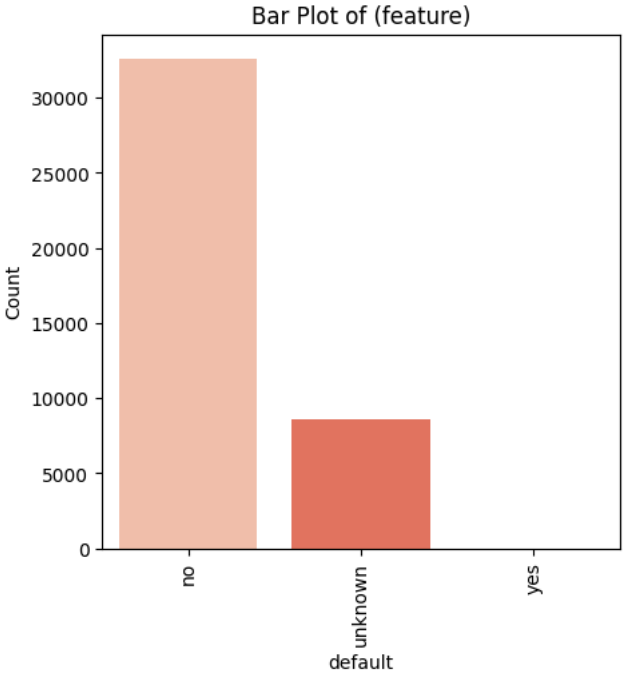
[4]

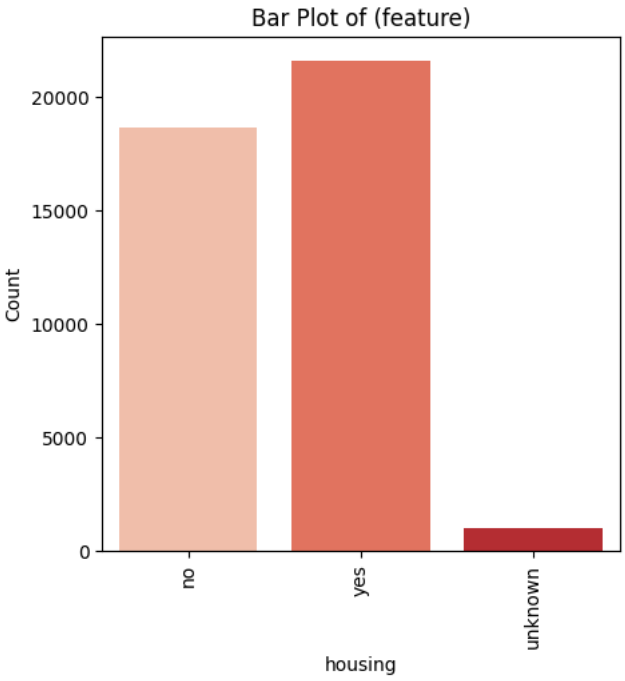

Bar Plot of (feature)
Count
0
2000
4000
6000
8000
10000
housemaid
services
admin.
blue-collar
technician
retired
management
unemployed
self-employed
unknown
entrepreneur
student
job

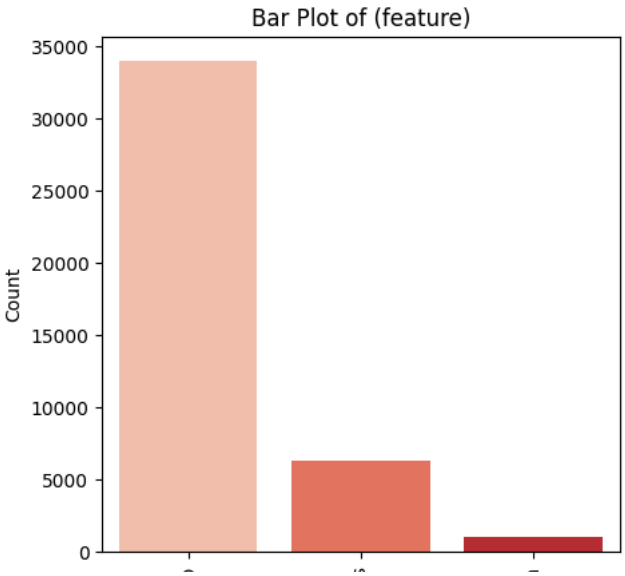

Bar Plot of (feature)
Count
0
5000
10000
15000
20000
25000
married
single
divorced
unknown
marital

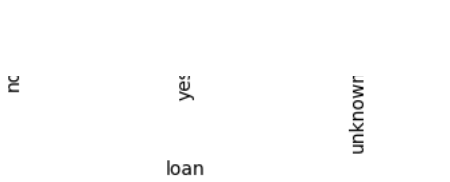

Bar Plot of (feature)
Count
0
2000
4000
6000
8000
10000
12000
basic.4y
basic.6y
basic.9y
unknown
illiterate

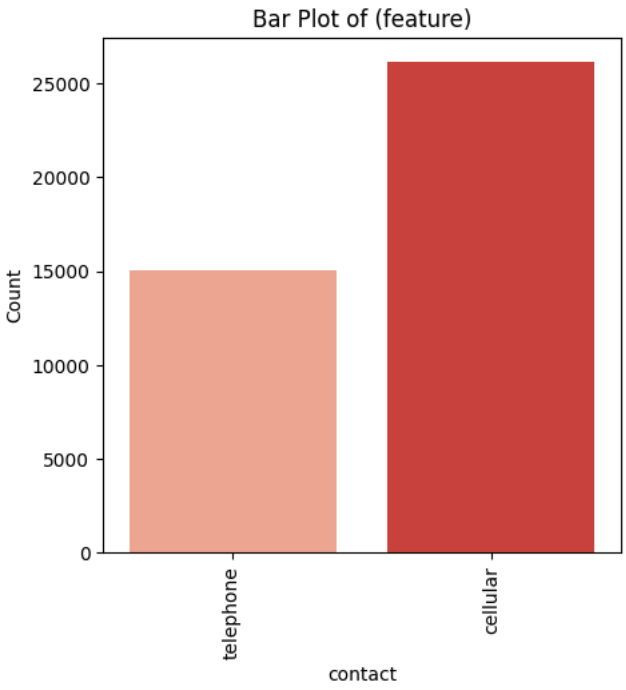

hig
professiona
universit
education

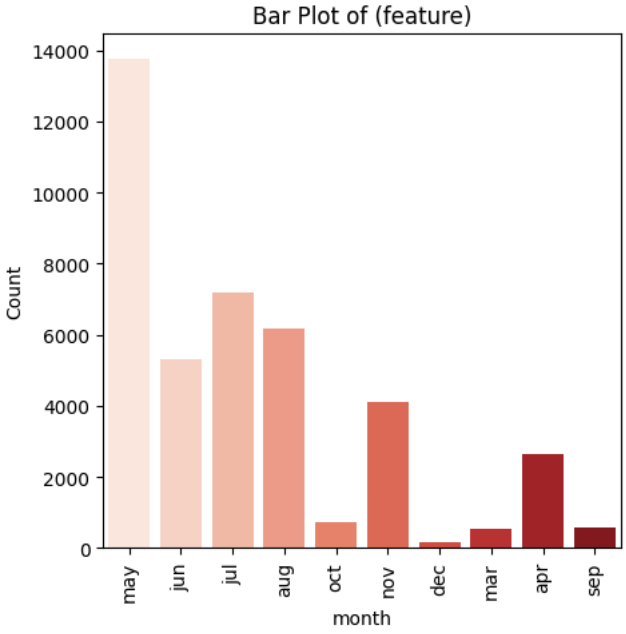

Bar Plot of (feature)
Count
30000
25000
20000
15000
10000
5000
0
no
unknown
yes
default

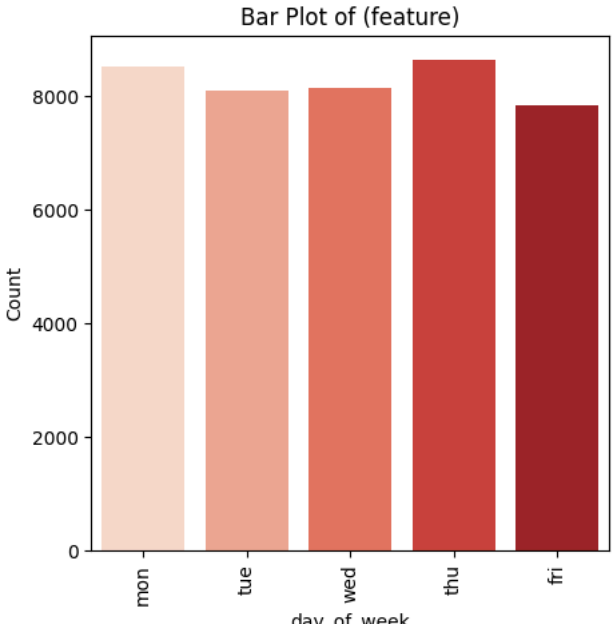

Bar Plot of (feature)
Count
20000
15000
10000
5000
0
no
yes
unknown
housing

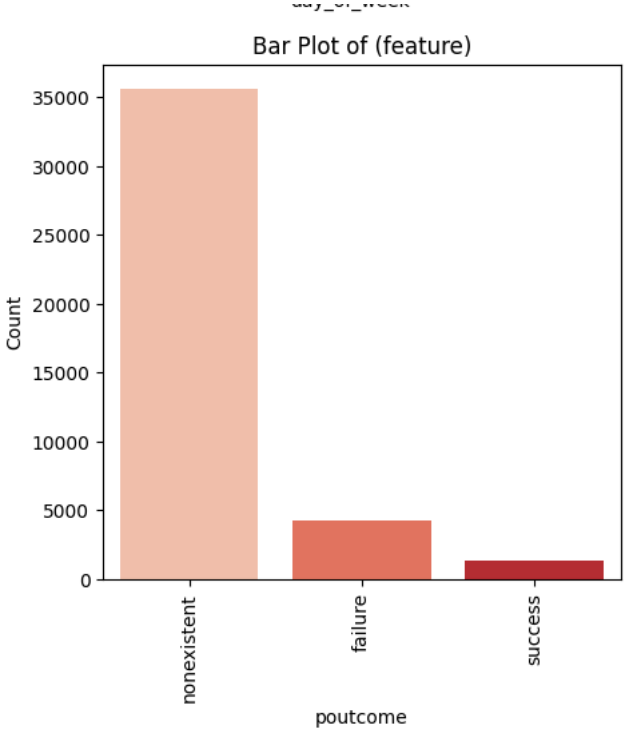

Bar Plot of (feature)
Count
35000
30000
25000
20000
15000
10000
5000
0

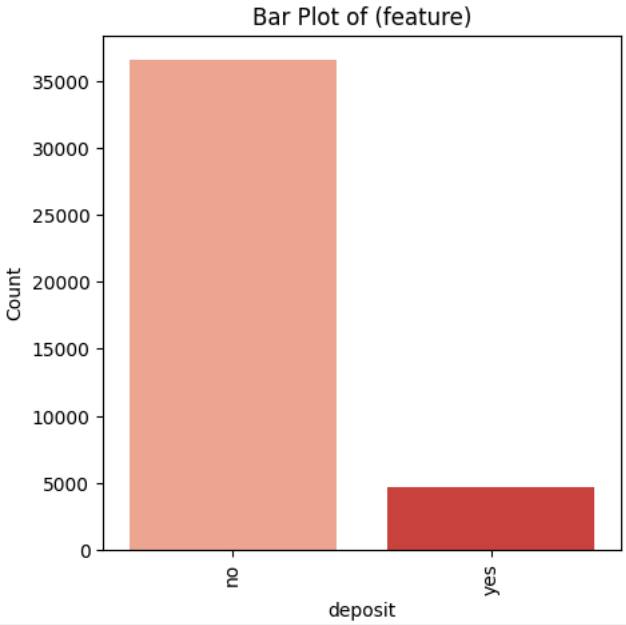

no
yes
unknown
loan


Bar Plot of (feature)
Count
25000
20000
15000
10000
5000
0
telephone
cellular
contact

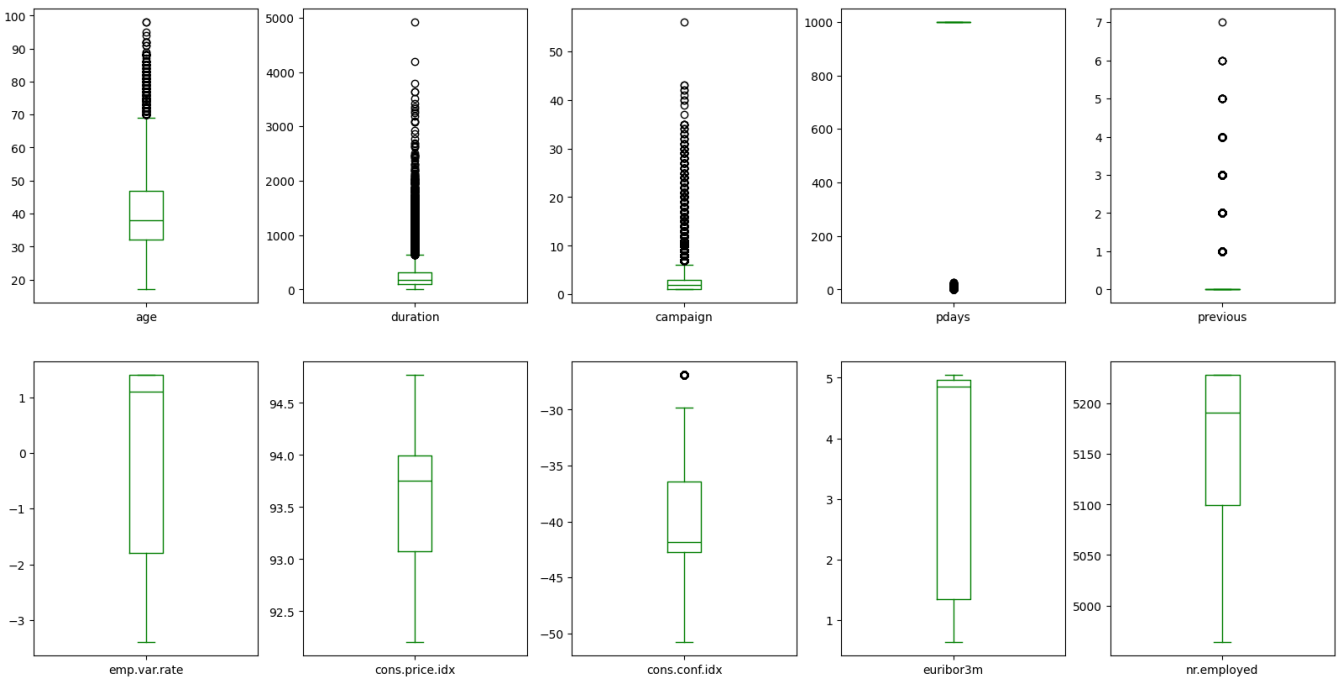

Bar Plot of (feature)
Count
14000
12000
10000
8000
6000
4000
2000
0
may
jun
jul
aug
oct
nov
dec
mar
apr
sep
month


Bar Plot of (feature)
Count
8000
6000
4000
2000
0
mon
tue
wed
thu
fri


Bar Plot of (feature)
Count
35000
30000
25000
20000
15000
10000
5000
0
nonexistent
failure
success
poutcome


Bar Plot of (feature)
Count
35000
30000
25000
20000
15000
10000
5000
0
no
yes
deposit

```
df.plot(kind='box',subplots=True,layout=(2,5),figsize=(20,10), color='green')
plt.show()
```




```
lb = LabelEncoder()
df1 = df.copy()
df_encoded = df1.apply(lb.fit_transform)
print(df_encoded)
```

```
       age  job  marital  education  default  housing  loan  contact  month  \
0       39    3        1          0        0        0     0        1      6
1       40    7        1          3        1        0     0        1      6
2       20    7        1          3        0        2     0        1      6
3       23    0        1          1        0        0     0        1      6
4       39    7        1          3        0        0     2        1      6
...    ...  ...      ...        ...      ...      ...   ...      ...    ...
41183   56    5        1          5        0        2     0        0      7
41184   29    1        1          5        0        0     0        0      7
41185   39    5        1          6        0        2     0        0      7
41186   27    9        1          5        0        0     0        0      7
41187   57    5        1          5        0        2     0        0      7

       day_of_week  ...  campaign  pdays  previous  poutcome  emp.var.rate  \
0                1  ...         0     26         0         1             8
1                1  ...         0     26         0         1             8
2                1  ...         0     26         0         1             8
3                1  ...         0     26         0         1             8
```