task33

August 14, 2024

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

[2]:
```
# Load the data
file_path = 'marketing-data.csv'
df = pd.read_csv(file_path)
```

[3]:
```
# Show the first few rows of the dataframe
print("First few rows of the dataset:")
print(df.head())
```

```
First few rows of the dataset:
   age           job  marital  education default  balance housing loan  \
0   58    management  married   tertiary      no     2143     yes   no
1   44    technician   single  secondary      no       29     yes   no
2   33  entrepreneur  married  secondary      no        2     yes  yes
3   47   blue-collar  married    unknown      no     1506     yes   no
4   33       unknown   single    unknown      no        1      no   no

   contact  day month  duration  campaign  pdays  previous poutcome is_success
0  unknown    5   may       261         1     -1         0  unknown         no
1  unknown    5   may       151         1     -1         0  unknown         no
2  unknown    5   may        76         1     -1         0  unknown         no
3  unknown    5   may        92         1     -1         0  unknown         no
4  unknown    5   may       198         1     -1         0  unknown         no
```

[4]:
```
df.head()
```

[4]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	is_success
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no

```
0  unknown    5   may       261          1     -1         0  unknown          no
1  unknown    5   may       151          1     -1         0  unknown          no
2  unknown    5   may        76          1     -1         0  unknown          no
3  unknown    5   may        92          1     -1         0  unknown          no
4  unknown    5   may       198          1     -1         0  unknown          no
```

[16]:
```
# Summary statistics
print("\nSummary statistics:")
# print(df.describe(include='all'))
df.describe(include='all')
```

```
Summary statistics:
```

[16]:

	age	job	marital	education	default	balance	\
count	45211.000000	45211	45211	45211	45211	45211.000000	
unique	NaN	12	3	4	2	NaN	
top	NaN	blue-collar	married	secondary	no	NaN	
freq	NaN	9732	27214	23202	44396	NaN	
mean	40.936210	NaN	NaN	NaN	NaN	1362.272058	
std	10.618762	NaN	NaN	NaN	NaN	3044.765829	
min	18.000000	NaN	NaN	NaN	NaN	-8019.000000	
25%	33.000000	NaN	NaN	NaN	NaN	72.000000	
50%	39.000000	NaN	NaN	NaN	NaN	448.000000	
75%	48.000000	NaN	NaN	NaN	NaN	1428.000000	
max	95.000000	NaN	NaN	NaN	NaN	102127.000000	

	housing	loan	contact	day	month	duration	\
count	45211	45211	45211	45211.000000	45211	45211.000000	
unique	2	2	3	NaN	12	NaN	
top	yes	no	cellular	NaN	may	NaN	
freq	25130	37967	29285	NaN	13766	NaN	
mean	NaN	NaN	NaN	15.806419	NaN	258.163080	
std	NaN	NaN	NaN	8.322476	NaN	257.527812	
min	NaN	NaN	NaN	1.000000	NaN	0.000000	
25%	NaN	NaN	NaN	8.000000	NaN	103.000000	
50%	NaN	NaN	NaN	16.000000	NaN	180.000000	
75%	NaN	NaN	NaN	21.000000	NaN	319.000000	
max	NaN	NaN	NaN	31.000000	NaN	4918.000000	

	campaign	pdays	previous	poutcome	is_success
count	45211.000000	45211.000000	45211.000000	45211	45211
unique	NaN	NaN	NaN	4	2
top	NaN	NaN	NaN	unknown	no
freq	NaN	NaN	NaN	36959	39922
mean	2.763841	40.197828	0.580323	NaN	NaN
std	3.098021	100.128746	2.303441	NaN	NaN

```
min          1.000000     -1.000000      0.000000      NaN         NaN
25%          1.000000     -1.000000      0.000000      NaN         NaN
50%          2.000000     -1.000000      0.000000      NaN         NaN
75%          3.000000     -1.000000      0.000000      NaN         NaN
max         63.000000    871.000000    275.000000      NaN         NaN
```

[6]:
```
# Check for missing values
print("\nMissing values:")
print(df.isnull().sum())
```

```
Missing values:
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
is_success    0
dtype: int64
```

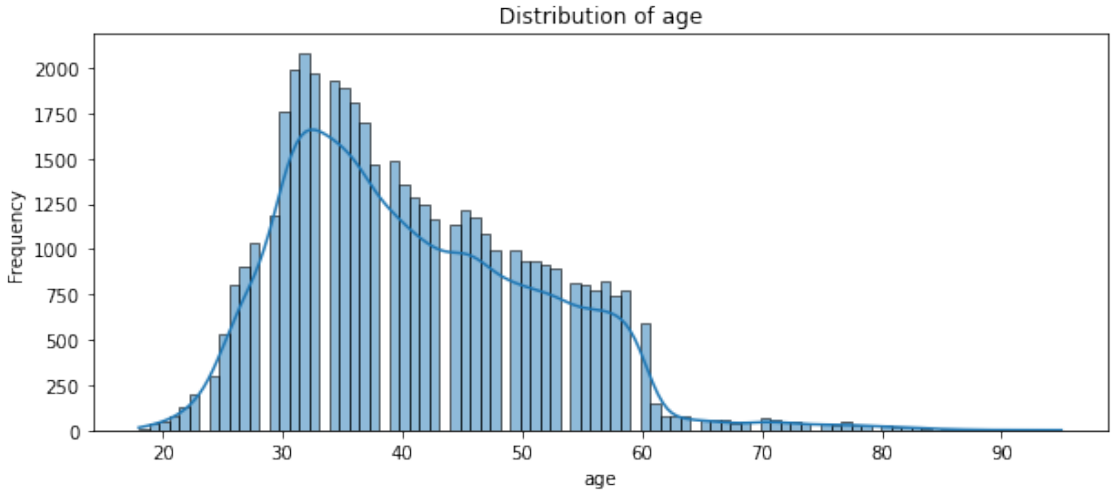
[7]:
```
# Data type info
print("\nData types:")
print(df.dtypes)
```

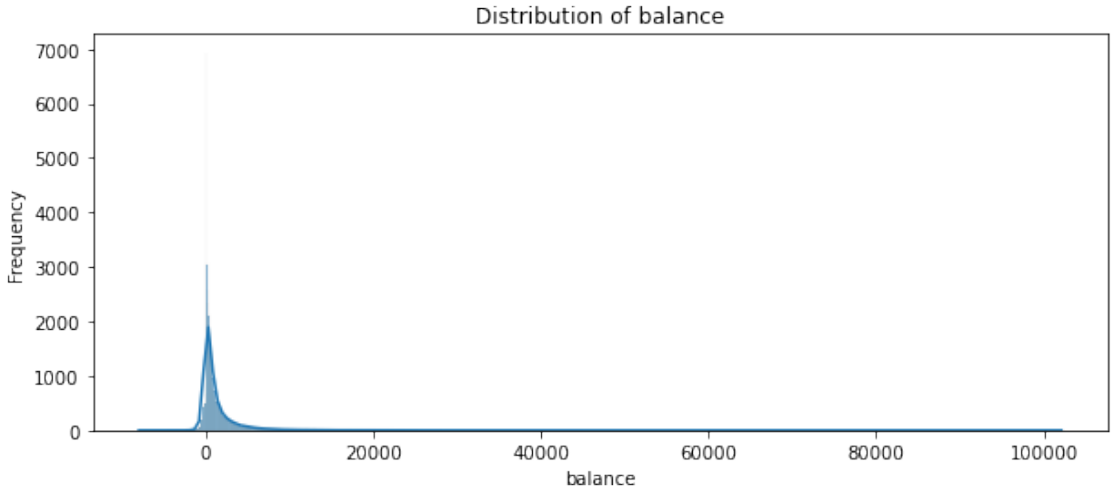
```
Data types:
age            int64
job           object
marital       object
education     object
default       object
balance        int64
housing       object
loan          object
contact       object
day            int64
month         object
```

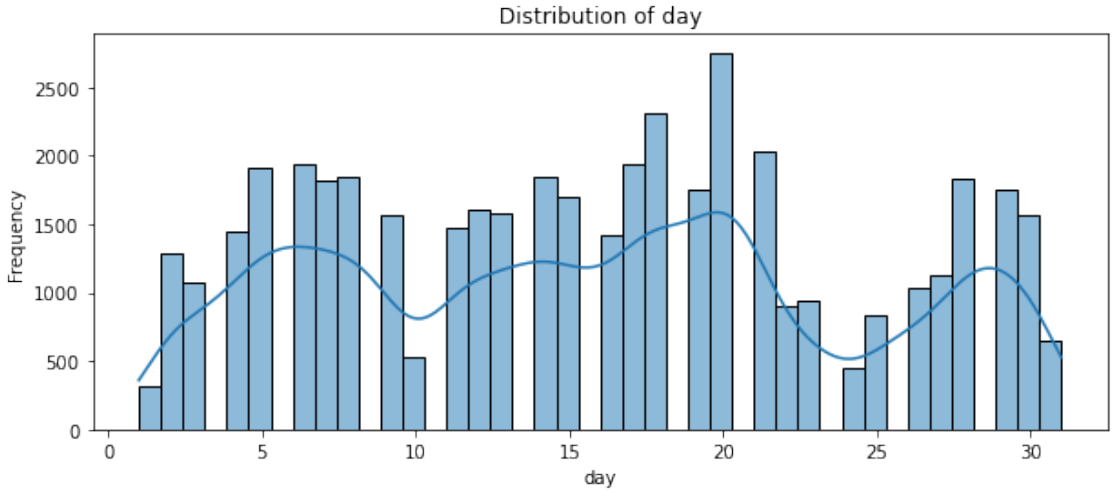
```
duration       int64
campaign       int64
pdays          int64
previous       int64
poutcome      object
is_success    object
dtype: object
```

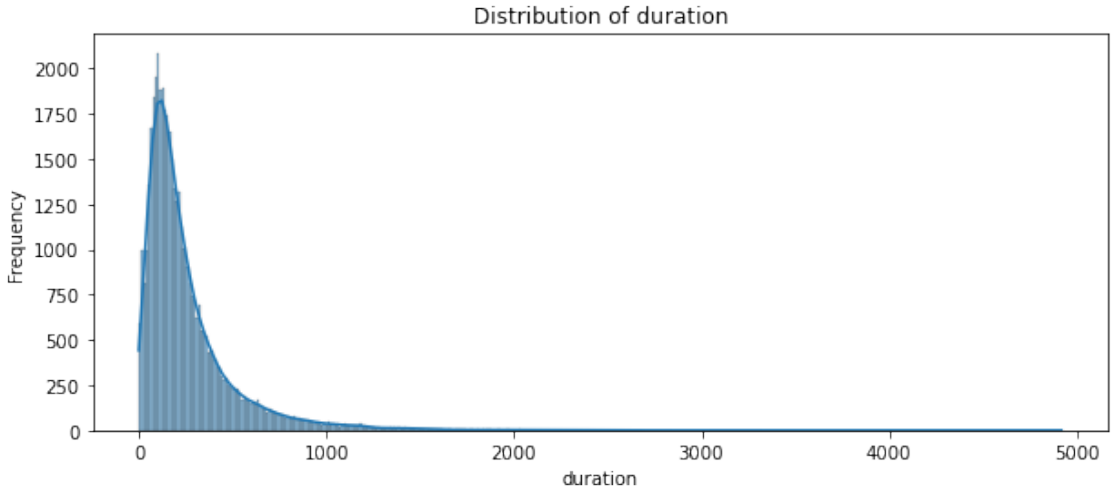
[8]:
```python
# Distribution of numerical variables
print("\nDistribution of numerical variables:")
numerical_cols = df.select_dtypes(include=np.number).columns
for col in numerical_cols:
    plt.figure(figsize=(10, 4))
    sns.histplot(df[col], kde=True)
    plt.title(f'Distribution of {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.show()
```

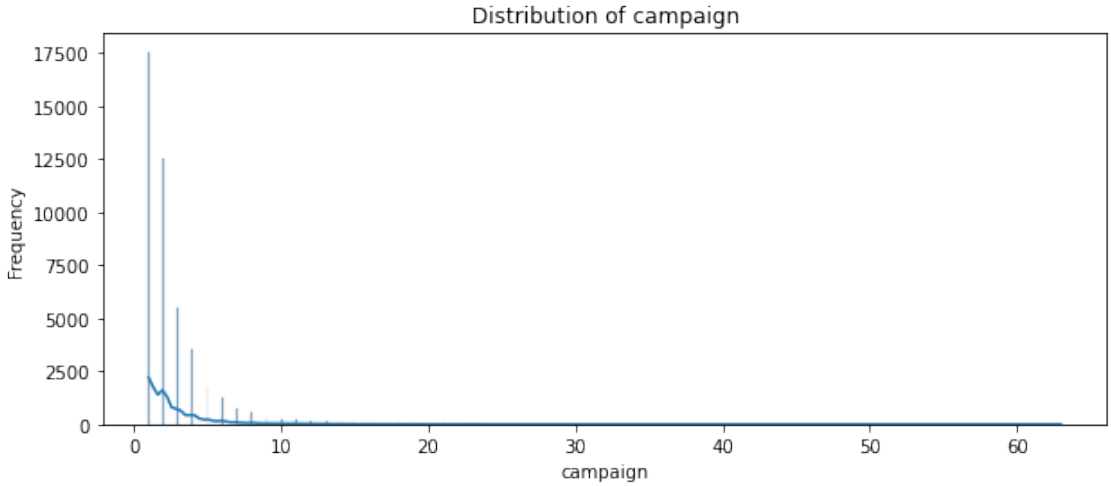
```
Distribution of numerical variables:
```

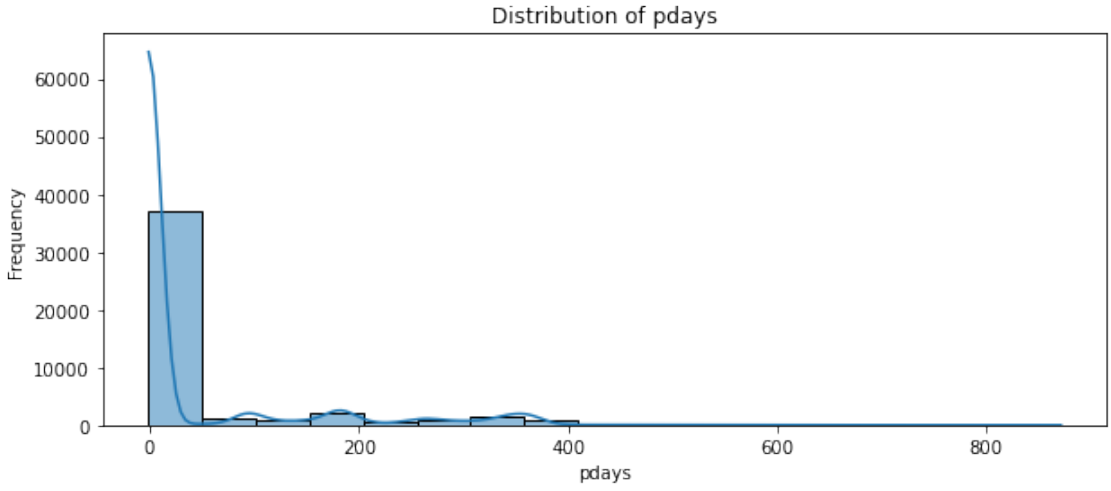


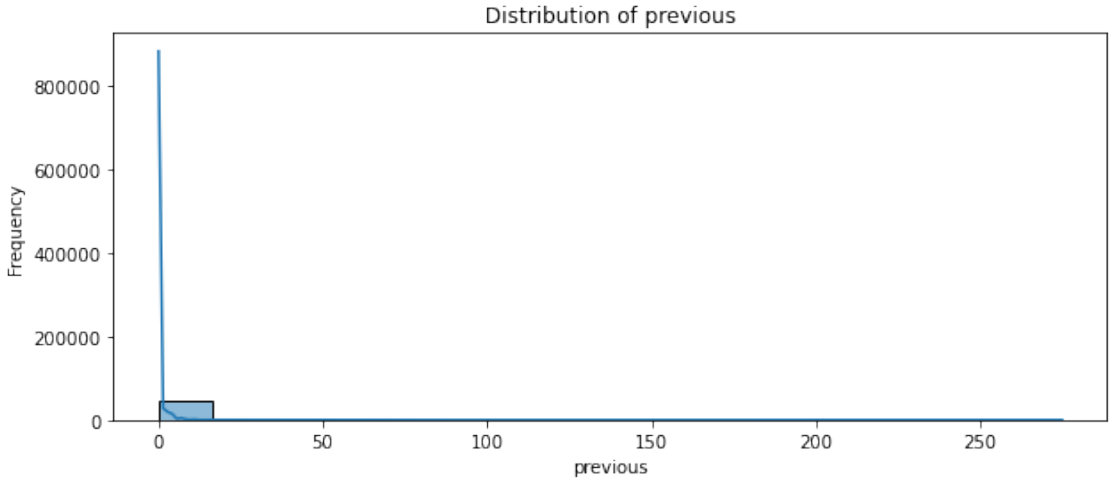

Distribution of balance
Frequency
balance
0
1000
2000
3000
4000
5000
6000
7000
0
20000
40000
60000
80000
100000


Distribution of day
Frequency
day
0
500
1000
1500
2000
2500
0
5
10
15
20
25
30


Distribution of duration
Frequency
duration
0
250
500
750
1000
1250
1500
1750
2000
0
1000
2000
3000
4000
5000

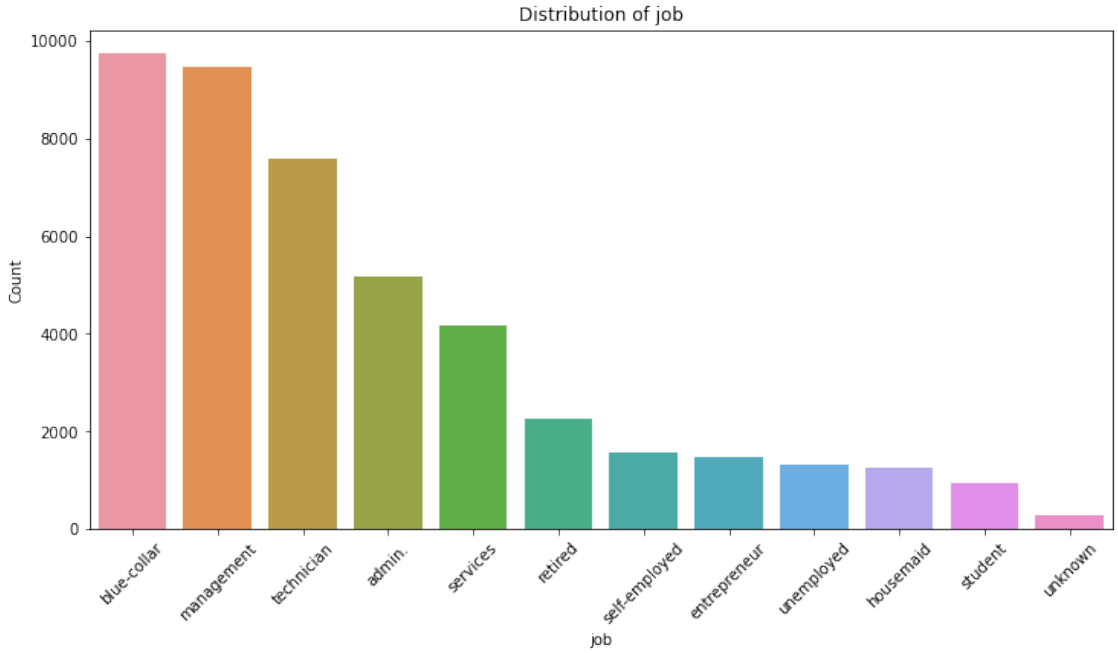

Distribution of campaign
Frequency
campaign
0
2500
5000
7500
10000
12500
15000
17500
0
10
20
30
40
50
60

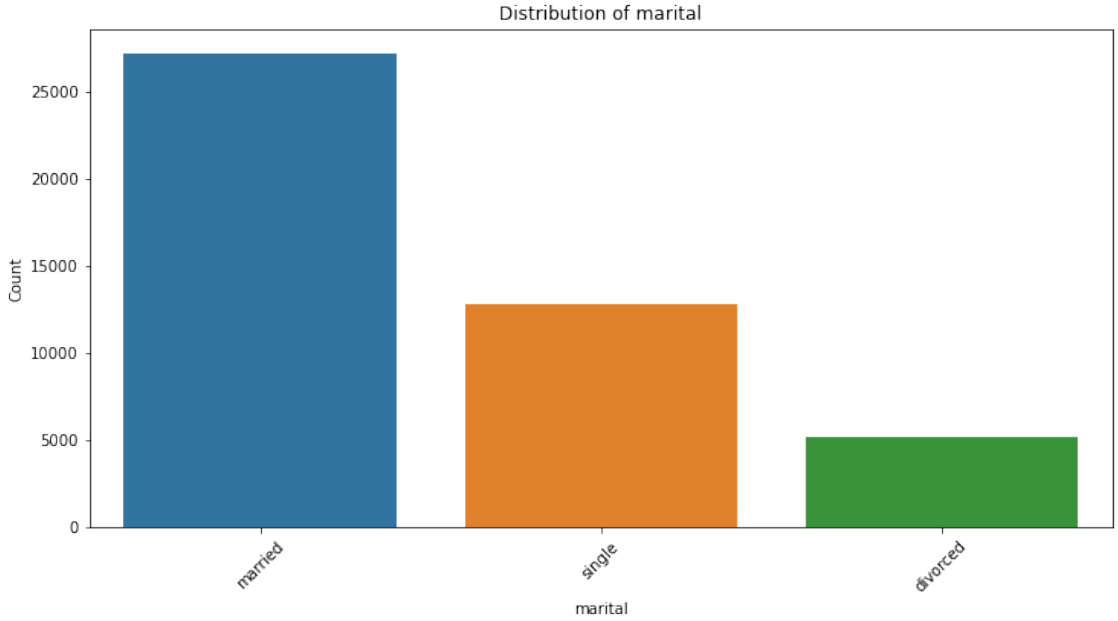

Distribution of pdays
Frequency
pdays
0
10000
20000
30000
40000
50000
60000
0
200
400
600
800

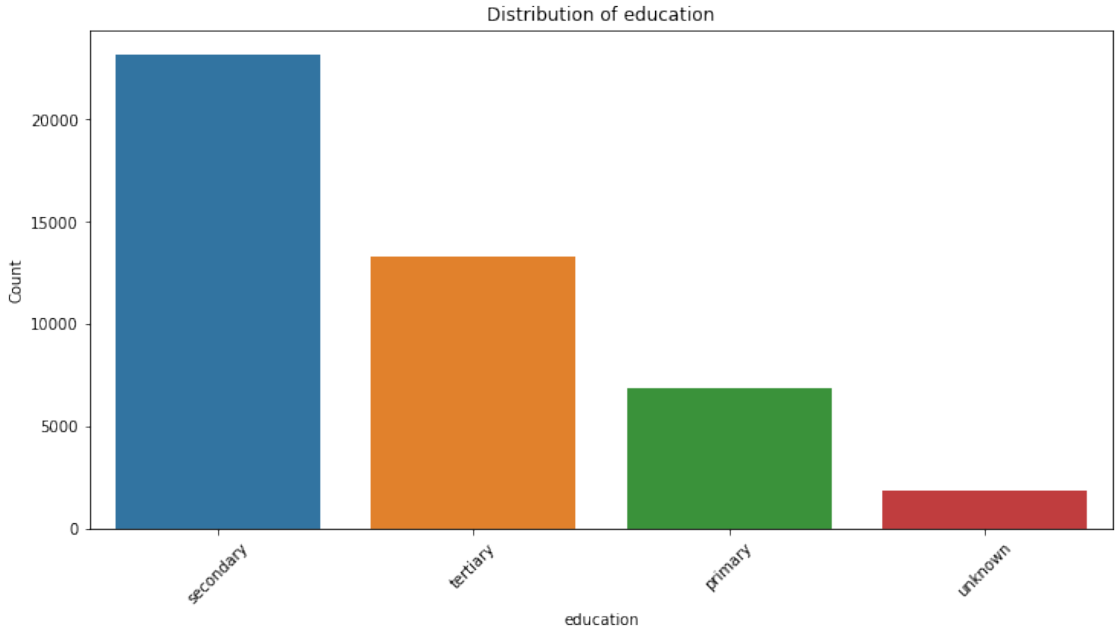


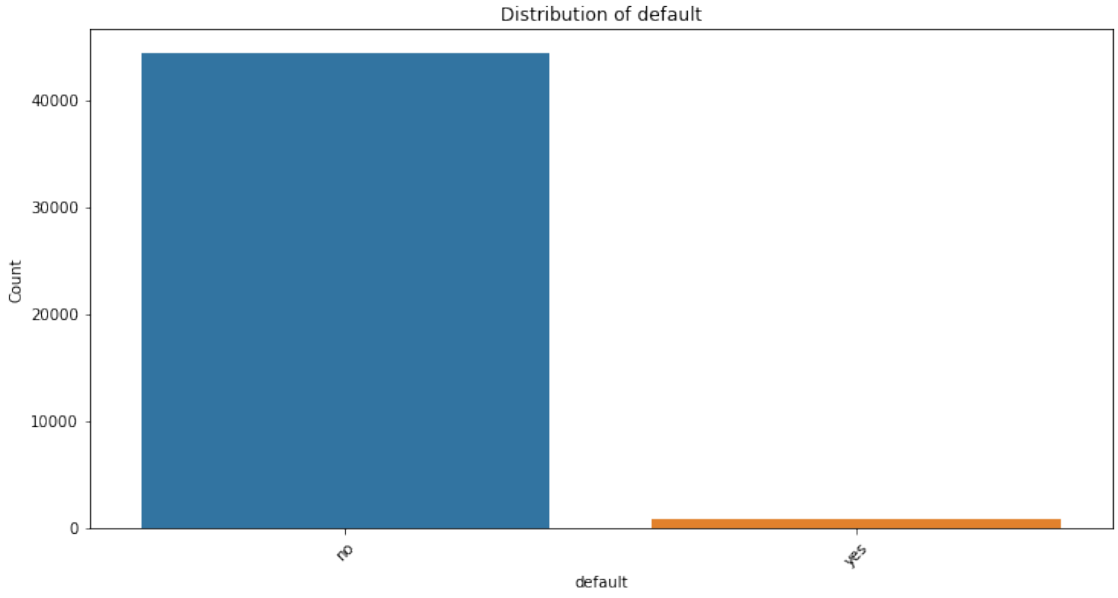
[10]:
```
# Distribution of categorical variables
print("\nDistribution of categorical variables:")
categorical_cols = df.select_dtypes(include='object').columns
for col in categorical_cols:
    plt.figure(figsize=(12, 6))
    sns.countplot(data=df, x=col, order=df[col].value_counts().index)
    plt.title(f'Distribution of {col}')
    plt.xlabel(col)
    plt.ylabel('Count')
    plt.xticks(rotation=45)
    plt.show()
```

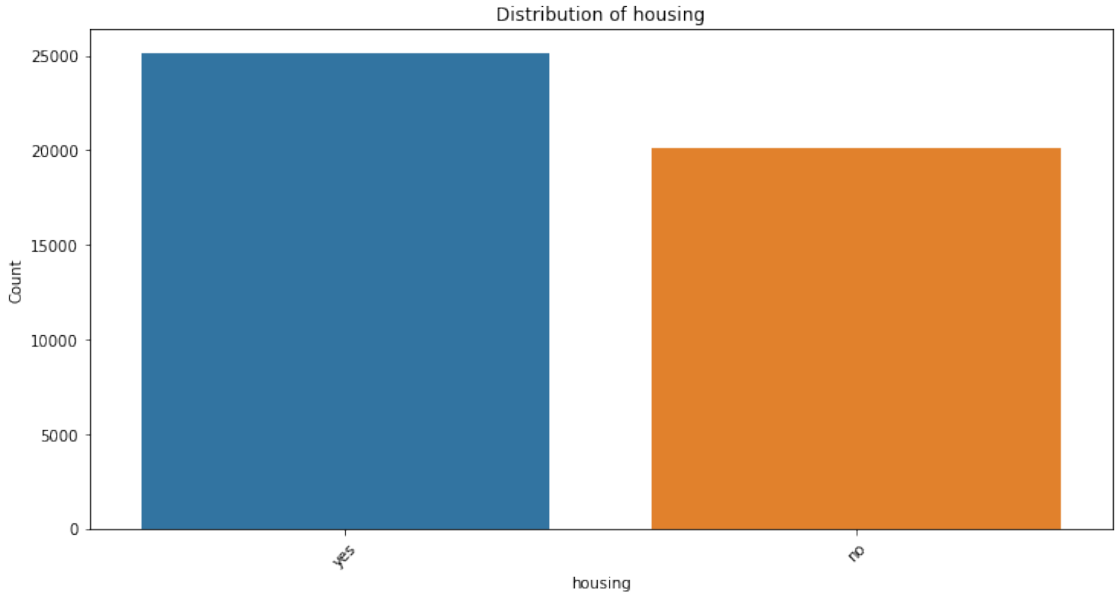
```
Distribution of categorical variables:
```

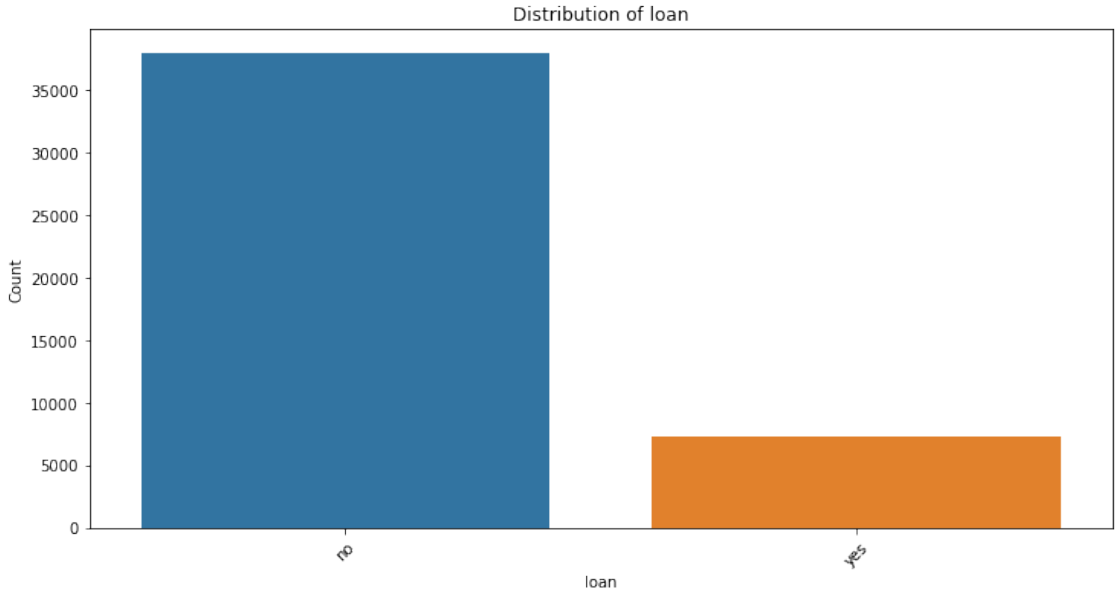

Distribution of job
Count
job
blue-collar
management
technician
admin.
services
retired
self-employed
entrepreneur
unemployed
housemaid
student
unknown

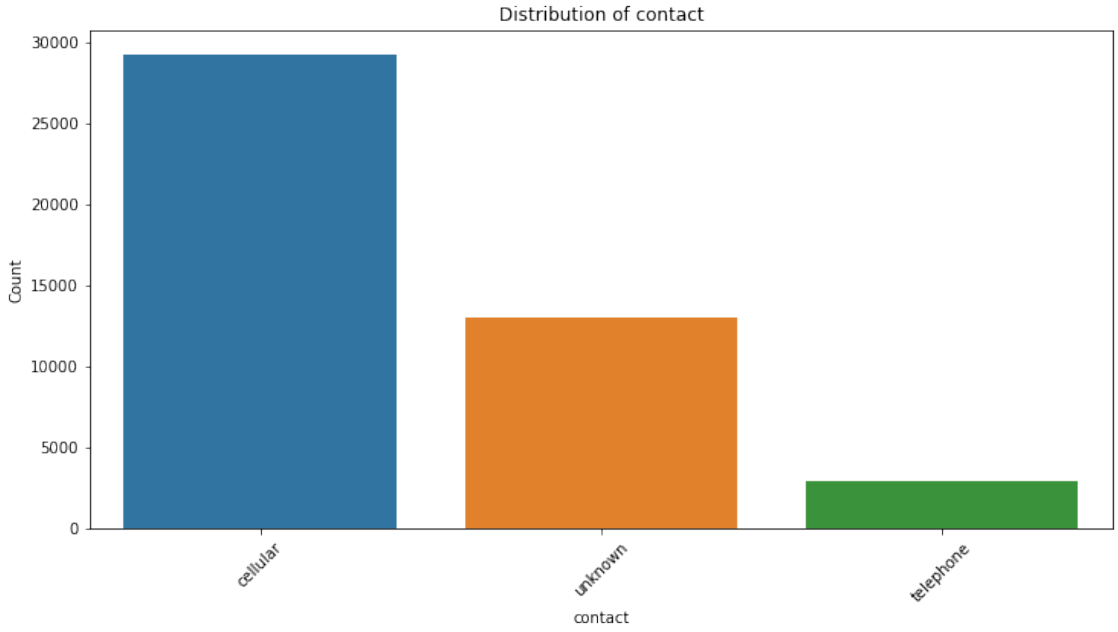

Distribution of marital
Count
marital
married
single
divorced

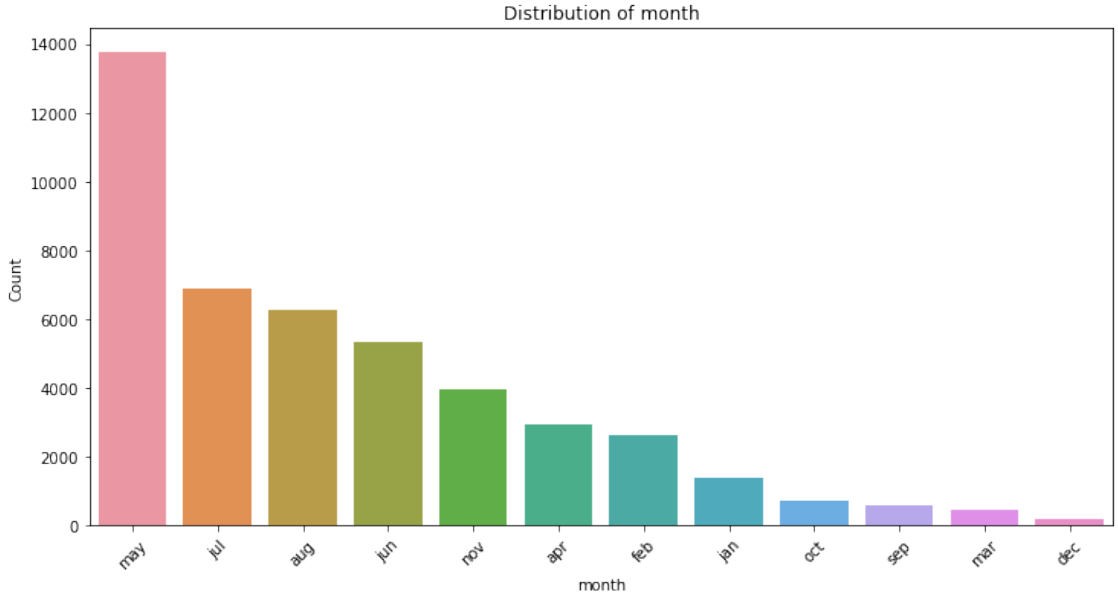

Distribution of education
Count
20000
15000
10000
5000
0
secondary
tertiary
primary
unknown
education

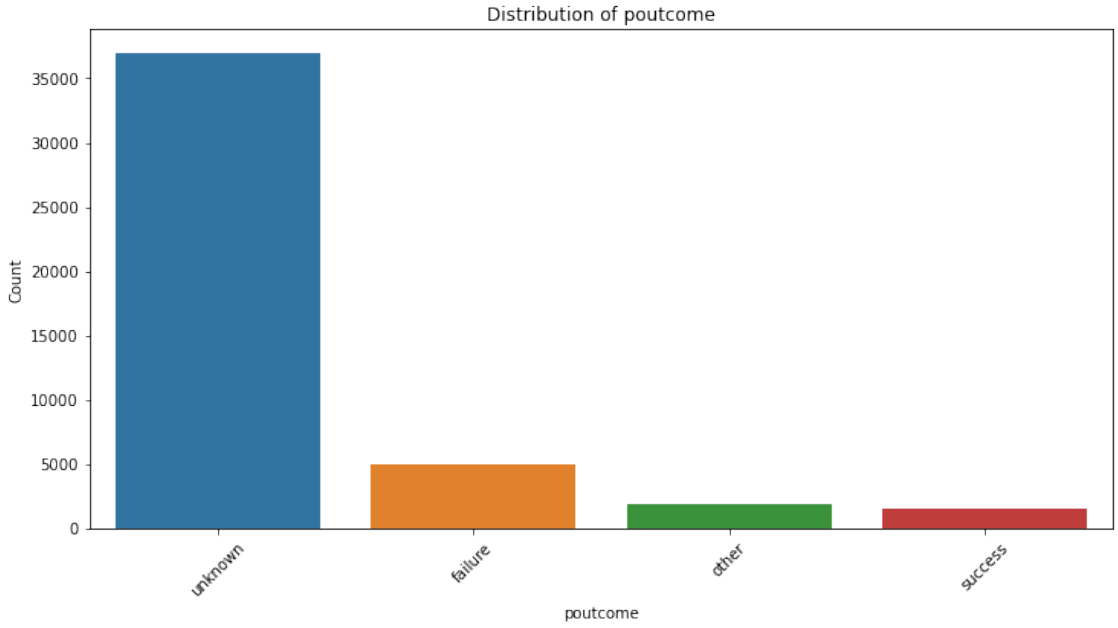

Distribution of default
Count
40000
30000
20000
10000
0
no
yes
default

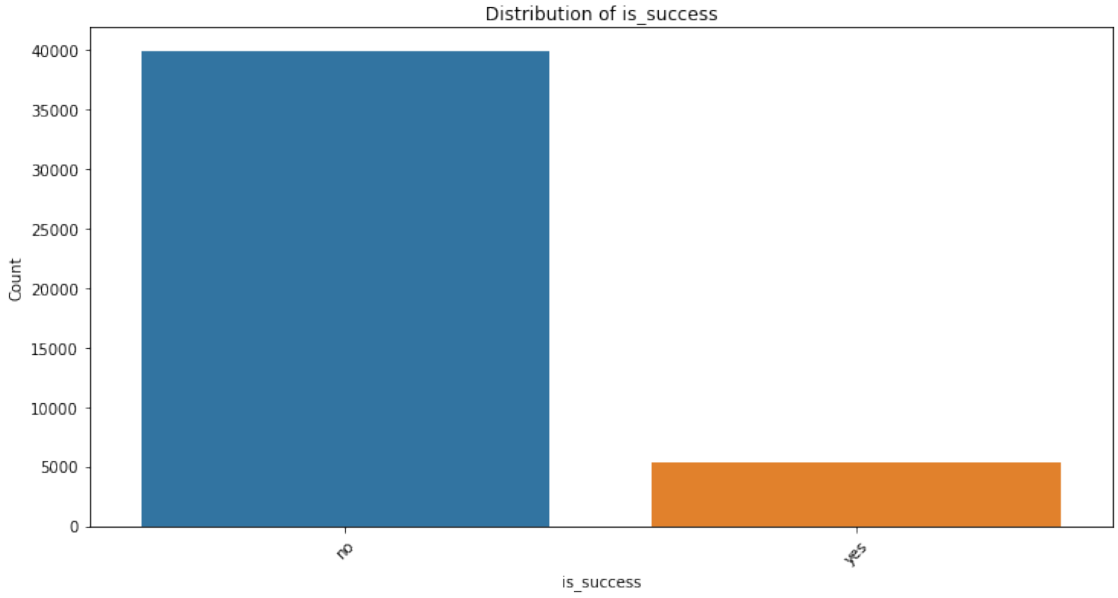

Distribution of housing
25000
20000
15000
10000
5000
0
Count
yes
no
housing


Distribution of loan
35000
30000
25000
20000
15000
10000
5000
0
Count
no
yes
loan


Distribution of contact
30000
25000
20000
15000
10000
5000
0
Count
cellular
unknown
telephone
contact


Distribution of month
14000
12000
10000
8000
6000
4000
2000
0
Count
may
jul
aug
jun
nov
apr
feb
jan
oct
sep
mar
dec
month





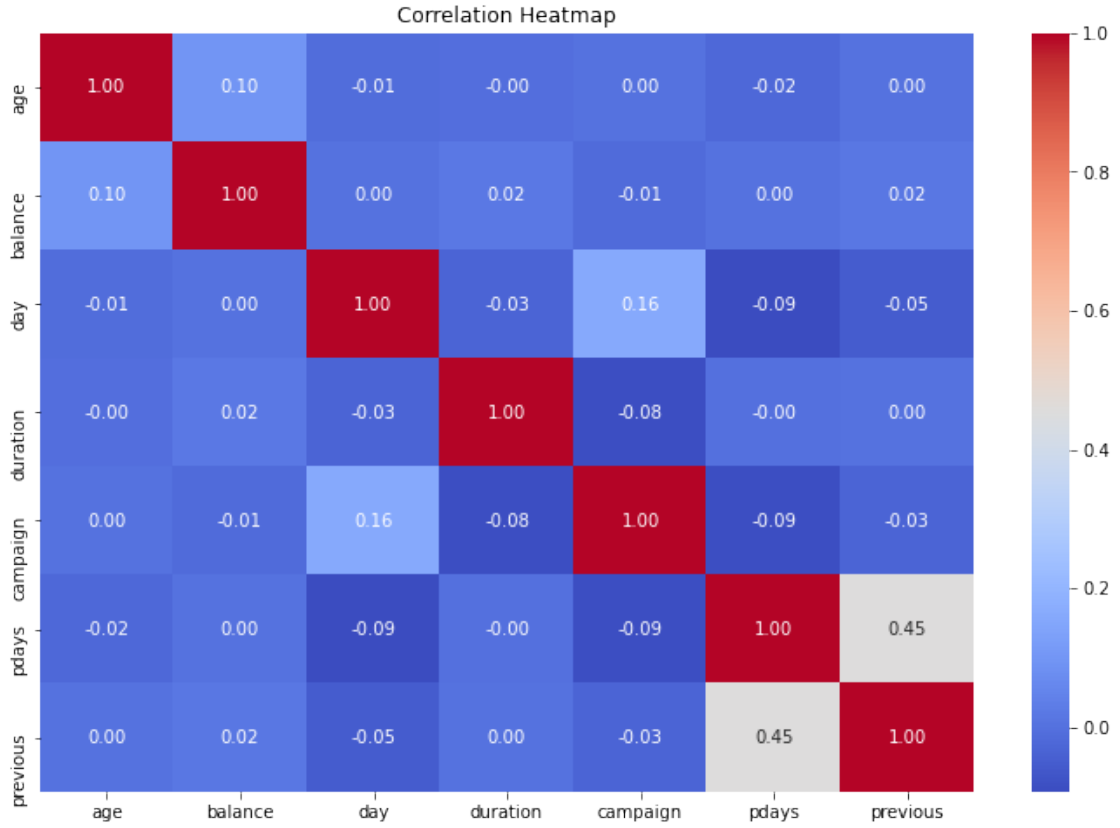
[14]:
```
# Analyze correlations between numerical features and the target variable
print("\nCorrelation matrix:")
correlation_matrix = df.corr()
correlation_matrix
```

```
Correlation matrix:
```

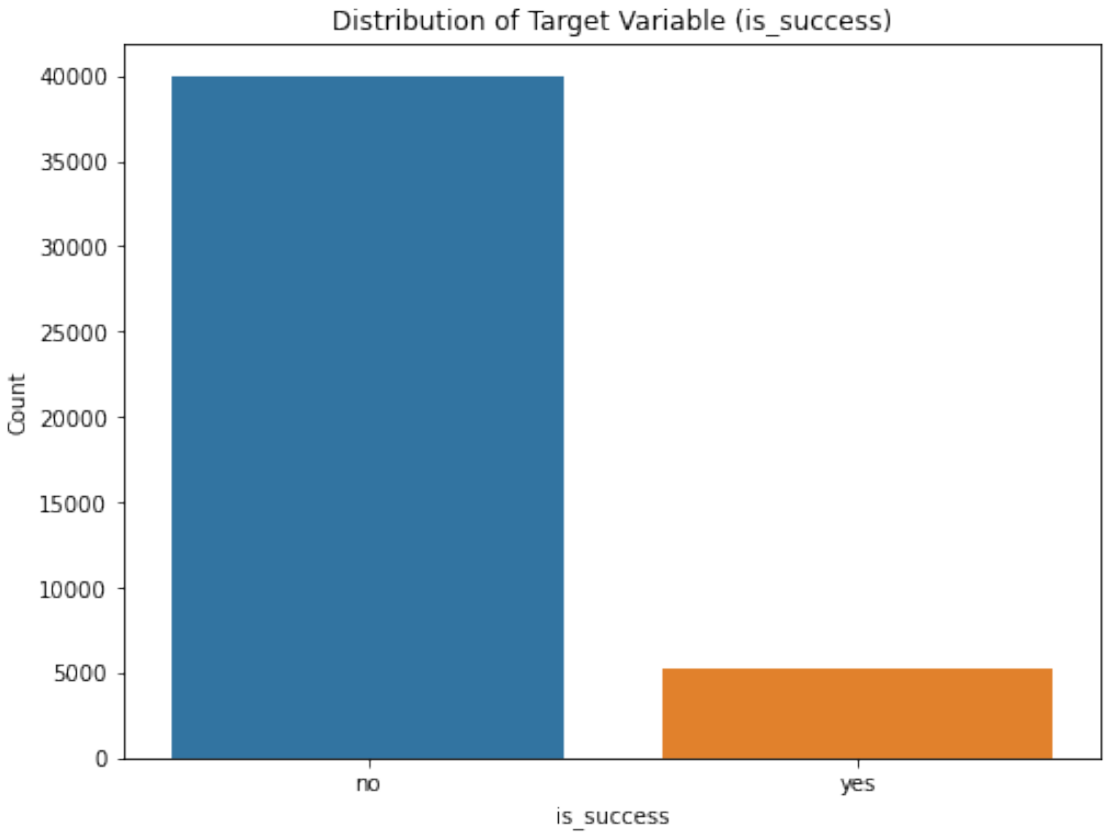
[14]:

	age	balance	day	duration	campaign	pdays	previous
age	1.000000	0.097783	-0.009120	-0.004648	0.004760	-0.023758	0.001288
balance	0.097783	1.000000	0.004503	0.021560	-0.014578	0.003435	0.016674
day	-0.009120	0.004503	1.000000	-0.030206	0.162490	-0.093044	-0.051710
duration	-0.004648	0.021560	-0.030206	1.000000	-0.084570	-0.001565	0.001203
campaign	0.004760	-0.014578	0.162490	-0.084570	1.000000	-0.088628	-0.032855
pdays	-0.023758	0.003435	-0.093044	-0.001565	-0.088628	1.000000	0.454820
previous	0.001288	0.016674	-0.051710	0.001203	-0.032855	0.454820	1.000000

[12]:
```
# Heatmap of correlations
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```



[13]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='is_success')
plt.title('Distribution of Target Variable (is_success)')
plt.xlabel('is_success')
plt.ylabel('Count')
plt.show()
```


Distribution of Target Variable (is_success)
Count
40000
35000
30000
25000
20000
15000
10000
5000
0
no
yes
is_success

[ ]: