Bank Marketing Data Analysis

This notebook presents an analysis of the `bankmarketing.csv` dataset, which includes data related to a bank's marketing campaigns. The main goal is to understand customer behavior and predict whether a client will subscribe to a term deposit.

```
import pandas as pd

# Load the dataset
df = pd.read_csv('bankmarketing.csv')

# Display the first few rows
df.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_o
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	
1	57	services	married	high.school	unknown	no	no	telephone	may	
2	37	services	married	high.school	no	yes	no	telephone	may	
3	40	admin.	married	basic.6y	no	no	no	telephone	may	
4	56	services	married	high.school	no	no	yes	telephone	may	

5 rows × 21 columns

```
# Check for missing values
print("Missing values:\n", df.isnull().sum())

# Check data types
print("\nData types:\n", df.dtypes)
```

```
Missing values:
 age               0
job                0
marital            0
education          0
default            0
housing            0
loan               0
contact            0
month              0
```

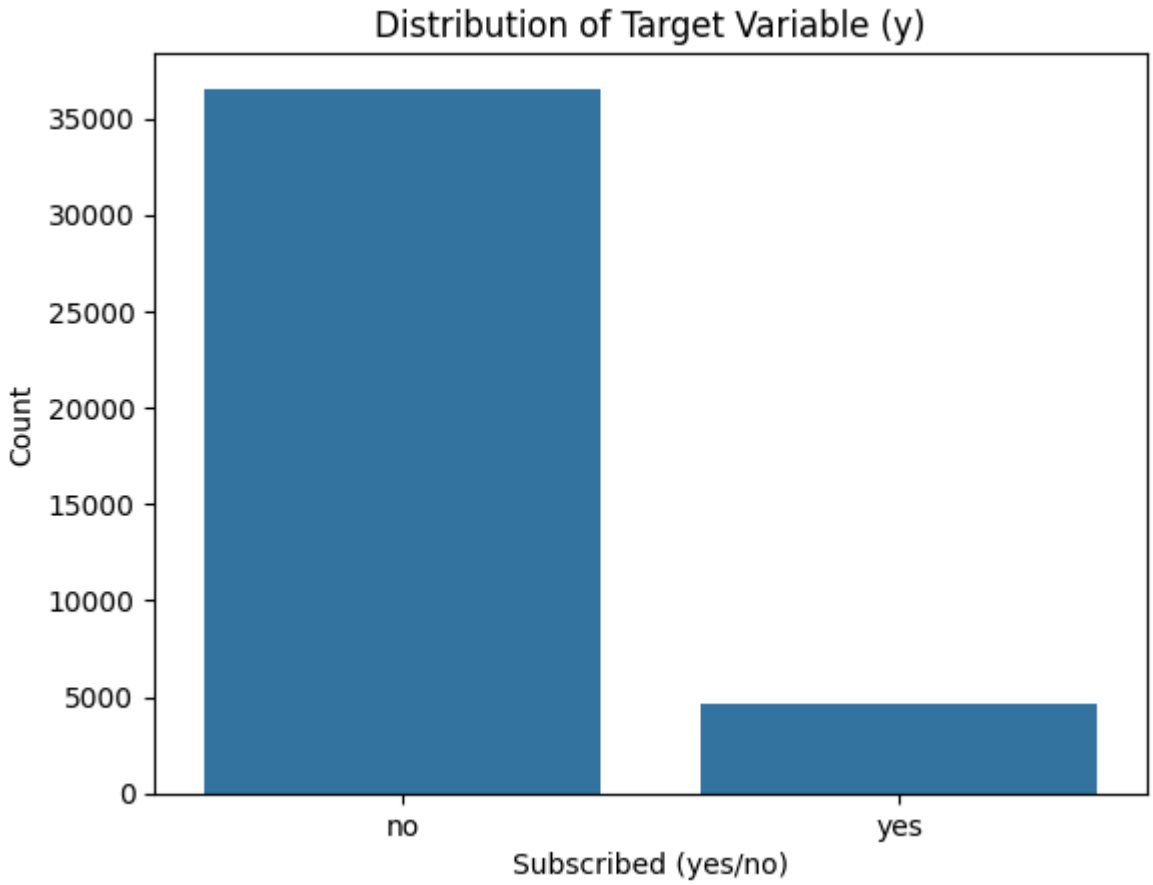
```
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
y                 0
dtype: int64

Data types:
 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
y                  object
dtype: object
```

```
# Summary statistics for numerical features
df.describe()
```

	age	duration	campaign	pdays	previous	emp.var.rate
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.00000(
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.08188(
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.57096(
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.40000(
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.80000(
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.10000(
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.40000(
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.40000(

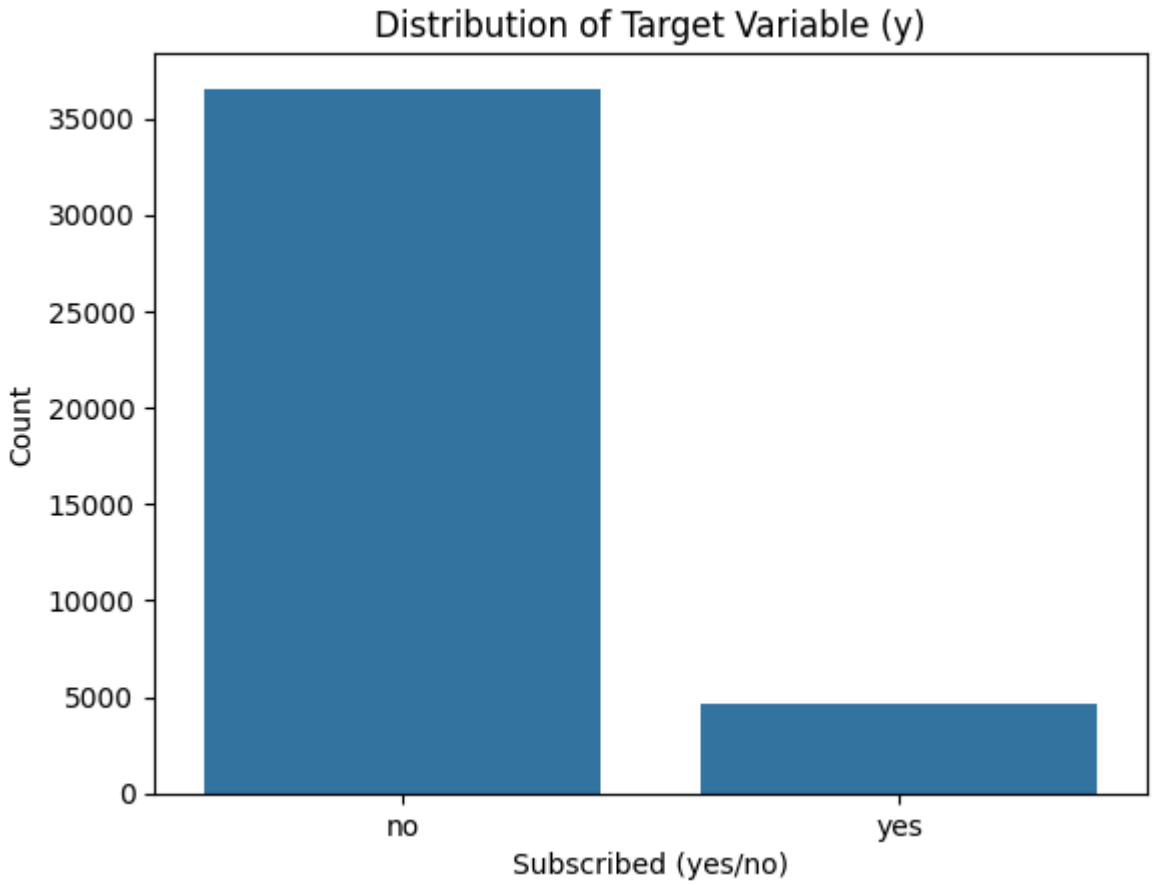
Start coding or generate with AI.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Plotting the count of target variable 'y'
sns.countplot(x='y', data=df)
plt.title('Distribution of Target Variable (y)')
plt.xlabel('Subscribed (yes/no)')
plt.ylabel('Count')
plt.show()
```



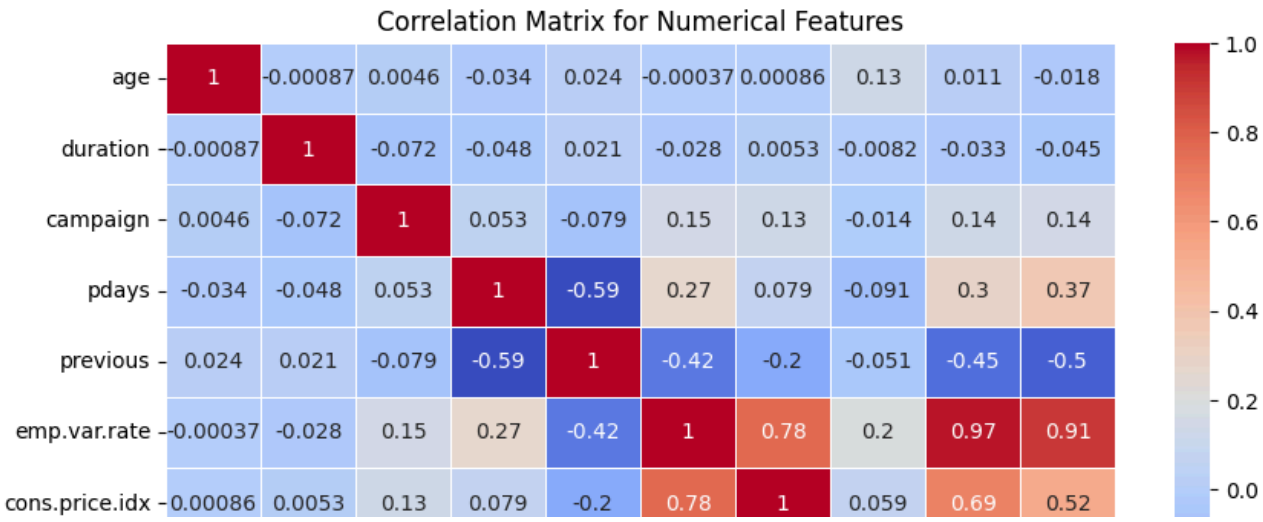
Double-click (or enter) to edit

```
import seaborn as sns
import matplotlib.pyplot as plt

# Plotting the count of target variable 'y'
sns.countplot(x='y', data=df)
plt.title('Distribution of Target Variable (y)')
plt.xlabel('Subscribed (yes/no)')
plt.ylabel('Count')
plt.show()
```



Double-click (or enter) to edit

```
# Correlation matrix for numerical features
import seaborn as sns
import matplotlib.pyplot as plt

# Select only numerical features
numerical_df = df.select_dtypes(include=['int64', 'float64'])

# Compute correlation matrix
corr_matrix = numerical_df.corr()

# Plot heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix for Numerical Features')
plt.show()
```


Correlation Matrix for Numerical Features
age
duration
campaign
pdays
previous
emp.var.rate
cons.price.idx
1 -0.00087 0.0046 -0.034 0.024 -0.00037 0.00086 0.13 0.011 -0.018
-0.00087 1 -0.072 -0.048 0.021 -0.028 0.0053 -0.0082 -0.033 -0.045
0.0046 -0.072 1 0.053 -0.079 0.15 0.13 -0.014 0.14 0.14
-0.034 -0.048 0.053 1 -0.59 0.27 0.079 -0.091 0.3 0.37
0.024 0.021 -0.079 -0.59 1 -0.42 -0.2 -0.051 -0.45 -0.5
-0.00037 -0.028 0.15 0.27 -0.42 1 0.78 0.2 0.97 0.91
0.00086 0.0053 0.13 0.079 -0.2 0.78 1 0.059 0.69 0.52
1.0
0.8
0.6
0.4
0.2
0.0