

```
y
['no' 'yes']
```

```
import numpy as np

# Identify features with missing values
features_na = [feature for feature in Bank_data.columns if Bank_data[feature].isnull().sum() > 0]

# Print the percentage of missing values for each feature with missing values
if features_na:
    for feature in features_na:
        print(f'{feature}: {np.round(Bank_data[feature].isnull().mean(), 4) * 100}% missing values')
else:
    print('No missing value found')
```

➦ No missing value found

find features with one value

```
for column in Bank_data.columns:
    print(column,Bank_data[column].nunique())
```

➦ sl. no 45211
age 77
job 12
marital 3
education 4
default 2
balance 7168
housing 2
loan 2
contact 3
day 31
month 12
duration 1573
campaign 48
pdays 559
previous 41
poutcome 4
y 2

Explore categorical features

```
categorical_features = [feature for feature in Bank_data.columns if (Bank_data[feature].dtypes == 'object') and (feature not in ['deposit'])]
categorical_features
```

➦ ['job',
'marital',
'education',
'default',
'housing',
'loan',
'contact',
'month',
'poutcome',
'y']

```
for feature in categorical_features:
    print('The feature is {} and the number of categories are {}'.format(feature, len(Bank_data[feature].unique())))
```

➦ The feature is job and the number of categories are 12
The feature is marital and the number of categories are 3
The feature is education and the number of categories are 4
The feature is default and the number of categories are 2
The feature is housing and the number of categories are 2
The feature is loan and the number of categories are 2
The feature is contact and the number of categories are 3
The feature is month and the number of categories are 12
The feature is poutcome and the number of categories are 4
The feature is y and the number of categories are 2

```
import warnings
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Suppress warnings
warnings.filterwarnings("ignore", category=FutureWarning)

# Ensure that the inline backend is used
%matplotlib inline

# Convert infinite values to NaN
Bank_data1 = Bank_data.replace([np.inf, -np.inf], np.nan)

# Generate pairplot
try:
    sns.pairplot(Bank_data1, hue='y', diag_kind='kde')
    plt.show()
except Exception as e:
    print("An error occurred while generating the pairplot:", e)
```



Data Cleaning

Handle Duplicates

Remove duplicates

```
Bank_data = Bank_data.drop_duplicates()
Bank_data
```

	sl. no	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	1	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	2	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	3	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	4	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	5	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
...
45206	45207	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	45208	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	45209	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	45210	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	45211	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

45211 rows x 18 columns

Handle missing values