```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report, ConfusionMatrixDisplay

df= pd.read_csv('bank.csv2.csv')
df.rename(columns={'y':'deposit'}, inplace=True)
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 4521,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 19,\n
\"max\": 87,\n        \"num_unique_values\": 67,\n        \"samples\":
[\n          50,\n          44,\n          36\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"job\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 12,\n
\"samples\": [\n          \"retired\",\n          \"housemaid\",\n
\"unemployed\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"married\",\n          \"single\",\n
\"divorced\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"education\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"secondary\",\n          \"unknown\",\n
\"primary\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"balance\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3009,\n
\"min\": -3313,\n        \"max\": 71188,\n
\"num_unique_values\": 2353,\n        \"samples\": [\n          1988,\
n          7010\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"housing\",\n      \"properties\": {\n        \"dtype\":
```

```
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"loan\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"contact\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"cellular\",\n
\"unknown\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"day\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 8,\n        \"min\": 1,\n        \"max\": 31,\n
\"num_unique_values\": 31,\n        \"samples\": [\n          10,\n
18\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"month\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 12,\n        \"samples\": [\n
\"mar\",\n          \"sep\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"duration\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 259,\n        \"min\": 4,\n
\"max\": 3025,\n        \"num_unique_values\": 875,\n
\"samples\": [\n          1516,\n          722\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"campaign\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3,\n
\"min\": 1,\n        \"max\": 50,\n        \"num_unique_values\": 32,\
n        \"samples\": [\n          28,\n          8\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"pdays\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 100,\n
\"min\": -1,\n        \"max\": 871,\n        \"num_unique_values\":
292,\n        \"samples\": [\n          63,\n          385\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"previous\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1,\n        \"min\": 0,\n
\"max\": 25,\n        \"num_unique_values\": 24,\n        \"samples\":
[\n          6,\n          11\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"poutcome\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"failure\",\n          \"success\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"deposit\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"yes\",\n          \"no\"\n
```

],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.head()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 4521,\n \"fields\": [\n   {\n   \"column\": \"age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 10,\n   \"min\": 19,\n   \"max\": 87,\n   \"num_unique_values\": 67,\n   \"samples\": [\n   50,\n   44,\n   36\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"job\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 12,\n   \"samples\": [\n   \"retired\",\n   \"housemaid\",\n   \"unemployed\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"marital\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 3,\n   \"samples\": [\n   \"married\",\n   \"single\",\n   \"divorced\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"education\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   \"secondary\",\n   \"unknown\",\n   \"primary\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"default\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"yes\",\n   \"no\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"balance\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 3009,\n   \"min\": -3313,\n   \"max\": 71188,\n   \"num_unique_values\": 2353,\n   \"samples\": [\n   1988,\n   7010\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"housing\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"yes\",\n   \"no\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"loan\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"yes\",\n   \"no\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"contact\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 3,\n   \"samples\": [\n   \"cellular\",\n   \"unknown\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"day\",\n   \"properties\": {\n   \"dtype\": \"number\",\n

\"std\": 8,\n   \"min\": 1,\n   \"max\": 31,\n
\"num_unique_values\": 31,\n   \"samples\": [\n   10,\n
18\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"month\",\n   \"properties\": {\n   \"dtype\": \"category\",\
n   \"num_unique_values\": 12,\n   \"samples\": [\n
\"mar\",\n   \"sep\"\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"duration\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 259,\n   \"min\": 4,\n
\"max\": 3025,\n   \"num_unique_values\": 875,\n
\"samples\": [\n   1516,\n   722\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"campaign\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 3,\n
\"min\": 1,\n   \"max\": 50,\n   \"num_unique_values\": 32,\
n   \"samples\": [\n   28,\n   8\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"pdays\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 100,\n
\"min\": -1,\n   \"max\": 871,\n   \"num_unique_values\":
292,\n   \"samples\": [\n   63,\n   385\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"previous\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 1,\n   \"min\": 0,\n
\"max\": 25,\n   \"num_unique_values\": 24,\n   \"samples\":
[\n   6,\n   11\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"poutcome\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 4,\n   \"samples\":
[\n   \"failure\",\n   \"success\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"deposit\",\n   \"properties\":
{\n   \"dtype\": \"category\",\n   \"num_unique_values\":
2,\n   \"samples\": [\n   \"yes\",\n   \"no\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.shape
```

```
(4521, 17)
```

```
df.tail()
```

{"type":"dataframe"}

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
```

```
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        4521 non-null   int64 
 1   job        4521 non-null   object
 2   marital    4521 non-null   object
 3   education  4521 non-null   object
 4   default    4521 non-null   object
 5   balance    4521 non-null   int64 
 6   housing    4521 non-null   object
 7   loan       4521 non-null   object
 8   contact    4521 non-null   object
 9   day        4521 non-null   int64 
 10  month      4521 non-null   object
 11  duration   4521 non-null   int64 
 12  campaign   4521 non-null   int64 
 13  pdays      4521 non-null   int64 
 14  previous   4521 non-null   int64 
 15  poutcome   4521 non-null   object
 16  deposit    4521 non-null   object
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
```

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

```
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
deposit      0
dtype: int64
```

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

```
0
```

```
df.dropna(inplace=True)
```

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

```
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
```

```
df.drop_duplicates(inplace=True)
```

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

```
0
```

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

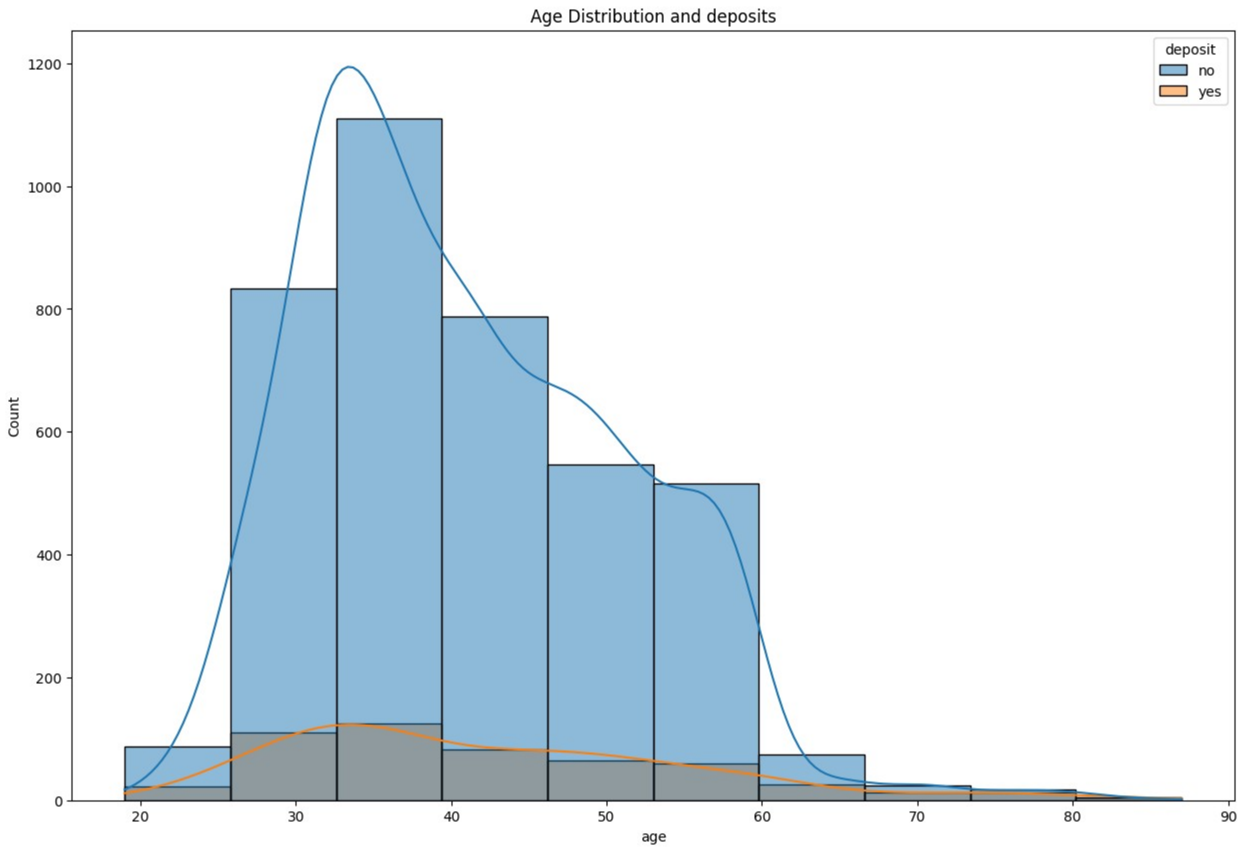
{"summary":"{\n \"name\": \"df\",\n \"rows\": 4,\n \"fields\": [\n   {\n   \"column\": \"job\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   12,\n   \"969\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"marital\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   3,\n   \"2797\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"education\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   4,\n   \"2306\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"default\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   2,\n   \"4445\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\

n   },\n   {\n   \"column\": \"housing\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   2,\n   \"2559\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"loan\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   2,\n   \"3830\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"contact\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   3,\n   \"2896\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"month\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   12,\n   \"1398\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"poutcome\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   4,\n   \"3705\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"deposit\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   2,\n   \"4000\",\n   \"4521\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

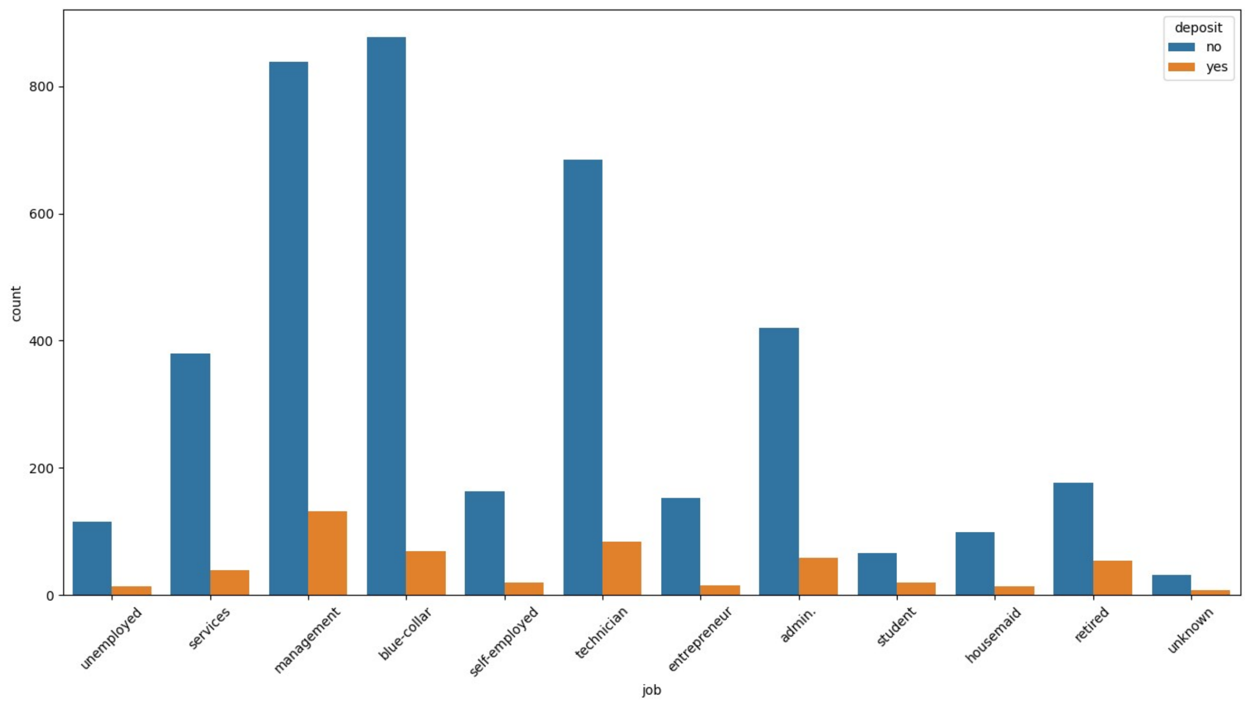
```
df.nunique()
```

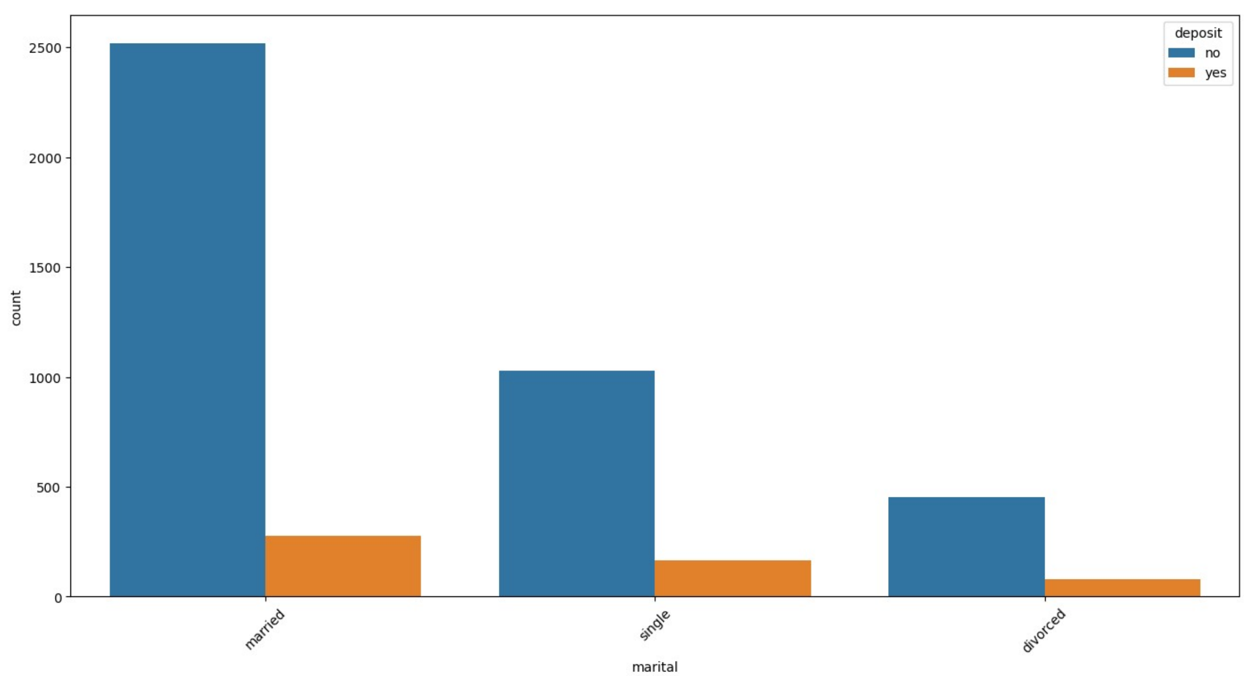
```
age            67
job            12
marital         3
education       4
default         2
balance      2353
housing         2
loan            2
contact         3
day            31
month          12
duration      875
campaign       32
pdays         292
previous       24
poutcome        4
deposit         2
dtype: int64
```

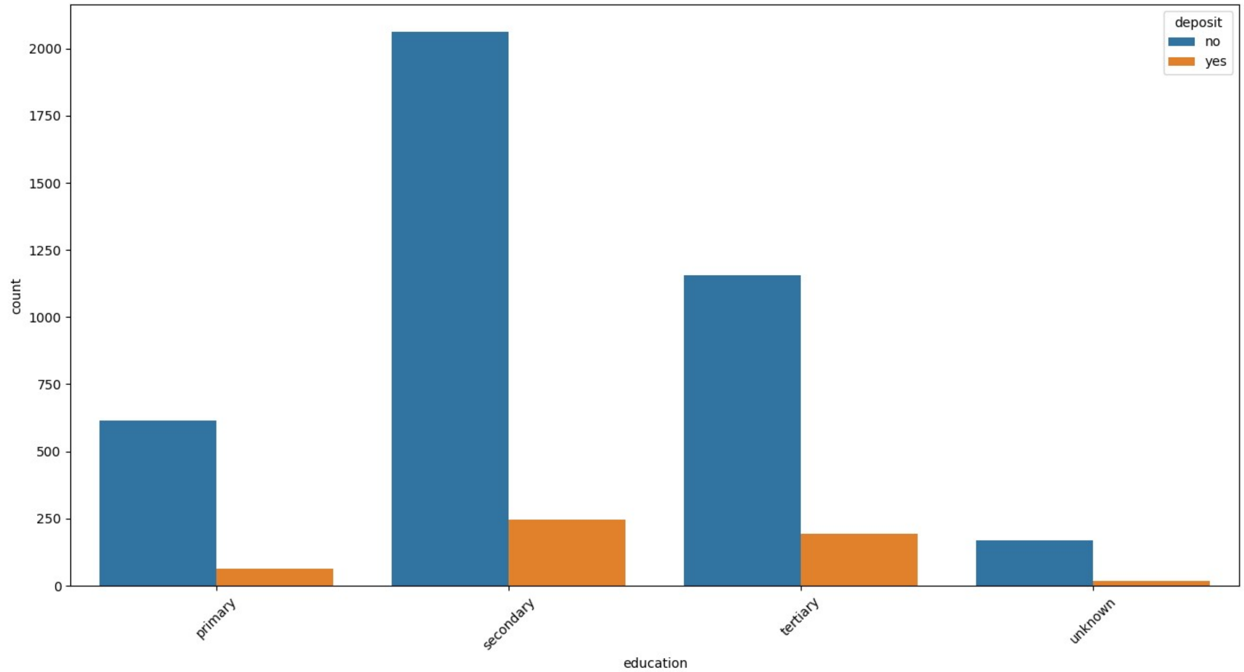
```python
#Exploratory data analysis
#Age distribution
plt.figure(figsize=(15,10))
sns.histplot(x='age', bins=10, kde=True, hue='deposit', data=df)
plt.title('Age Distribution and deposits')
plt.show()
```



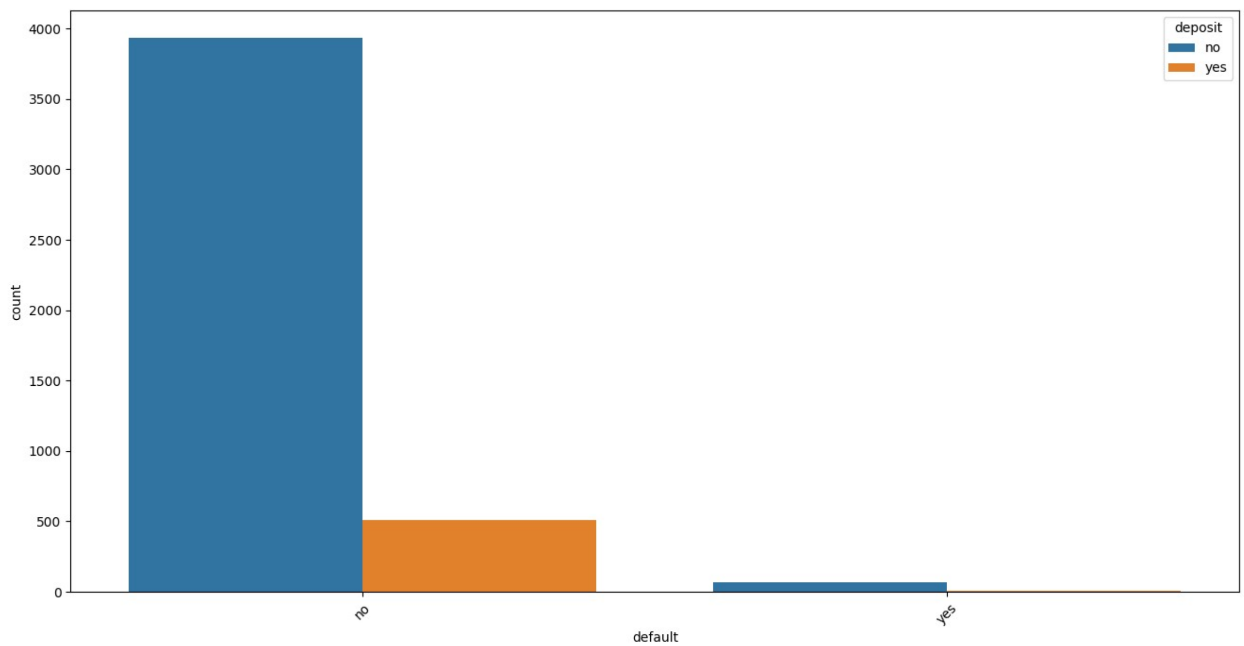

```python
cat= [i for i in df.columns if df[i].dtypes == 'object']
for i, features in enumerate(cat):
    plt.figure(figsize=(16,8))
    sns.countplot(x=features, hue='deposit', data=df)
    plt.xticks(rotation=45)
plt.show()
```


deposit
no
yes
count
0
200
400
600
800
unemployed
services
management
blue-collar
self-employed
technician
entrepreneur
admin.
student
housemaid
retired
unknown
job

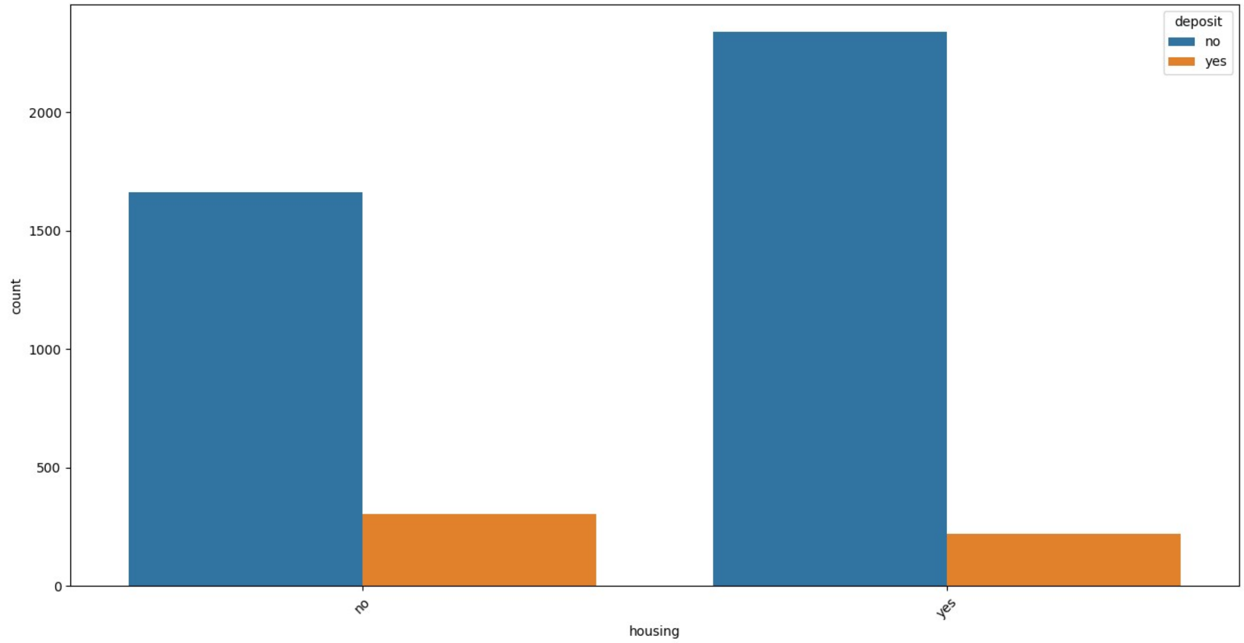

deposit
no
yes
count
0
500
1000
1500
2000
2500
married
single
divorced
marital

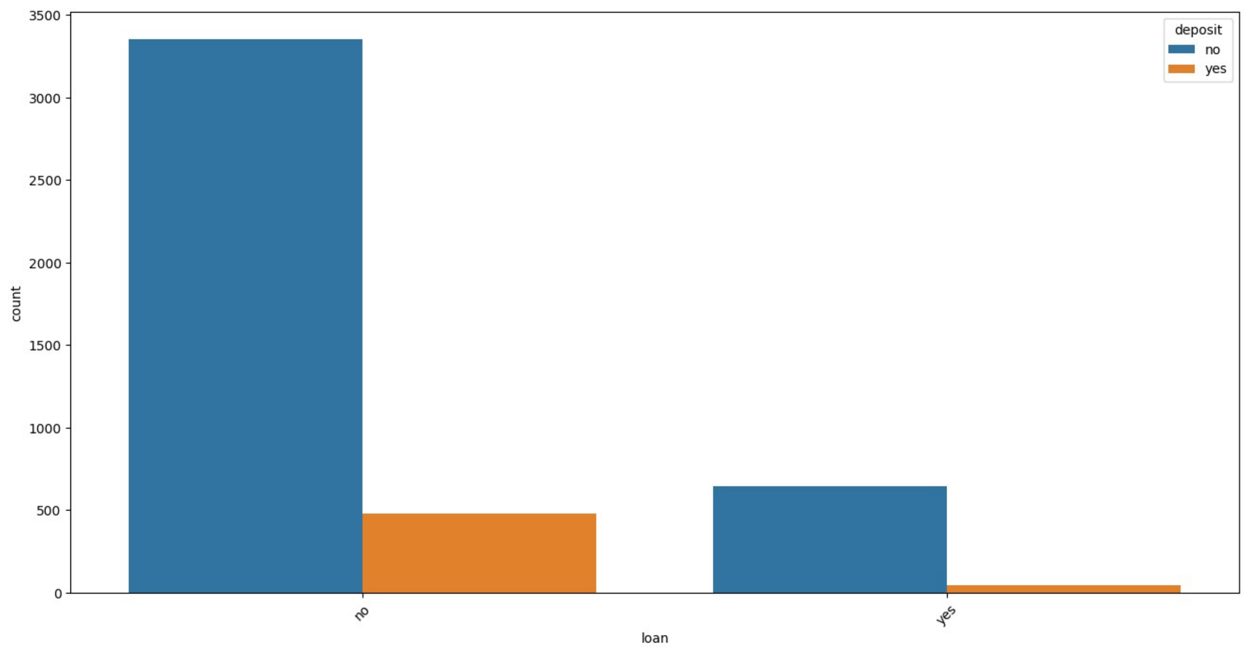


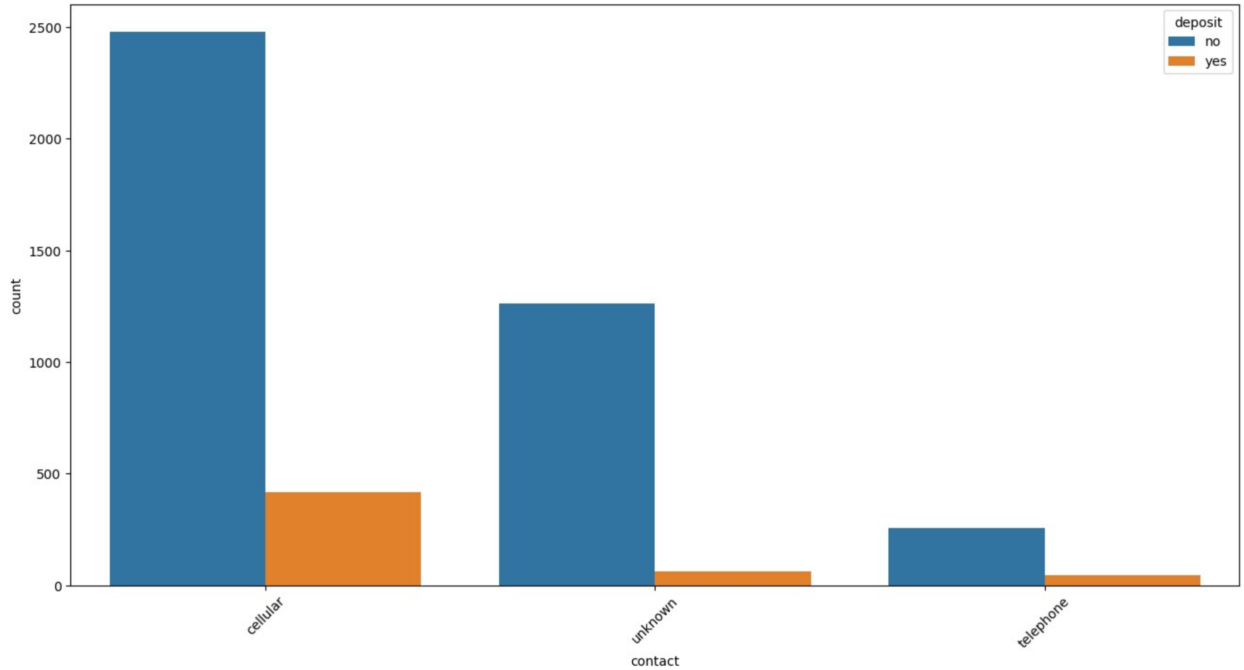
deposit
no
yes
count
2000
1750
1500
1250
1000
750
500
250
0
primary
secondary
tertiary
unknown
education

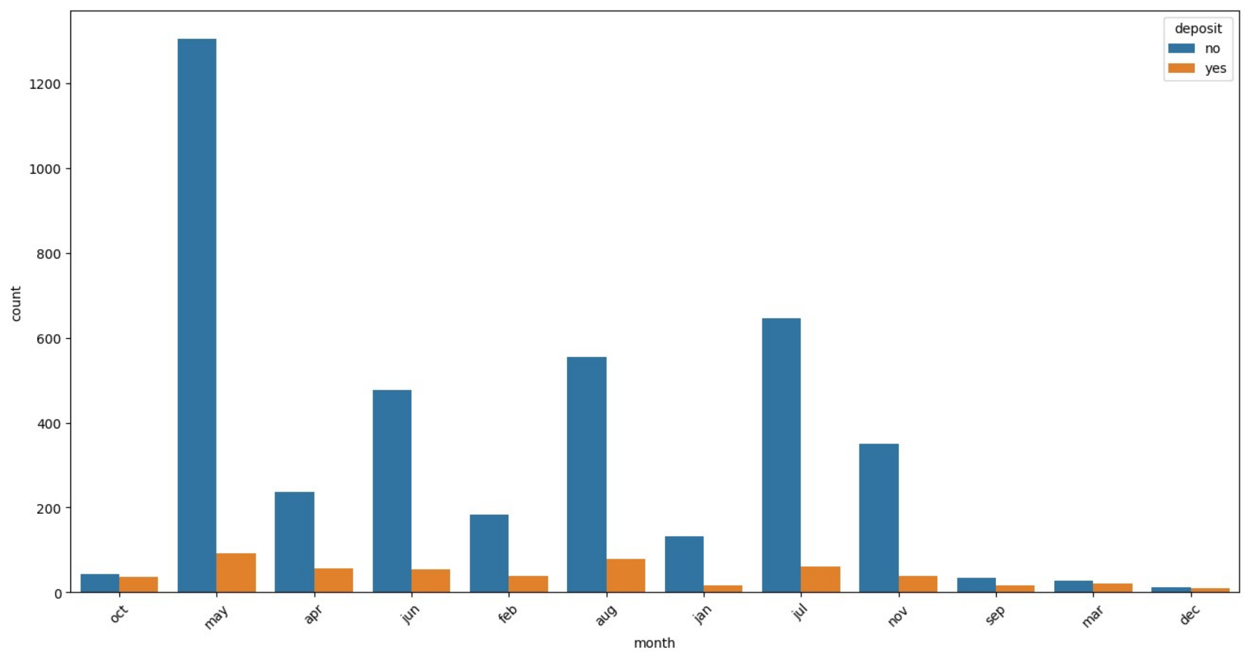


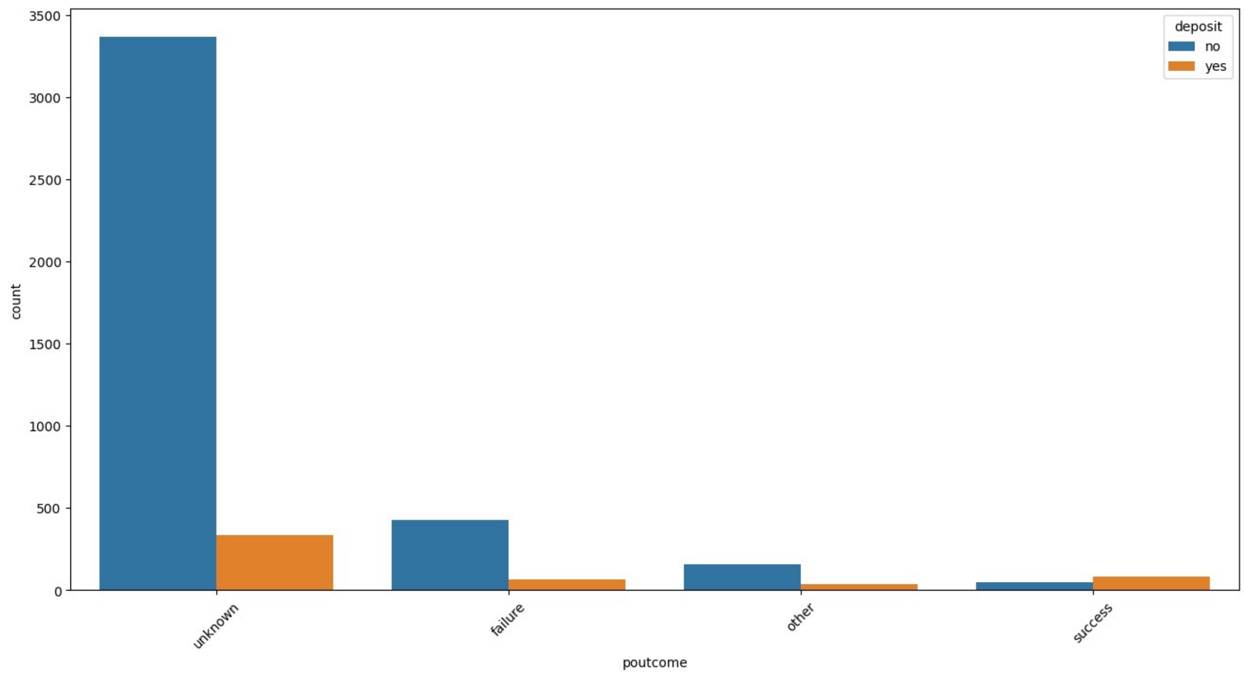
deposit
no
yes
count
4000
3500
3000
2500
2000
1500
1000
500
0
no
yes
default

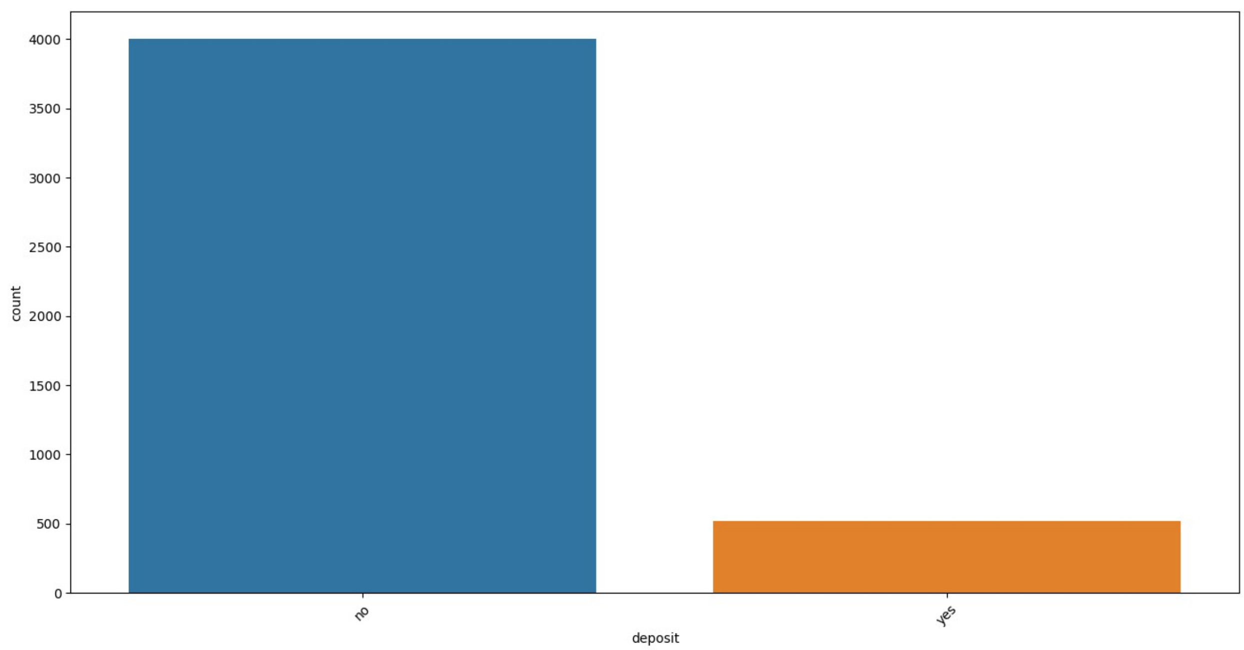

deposit
no
yes
2000
1500
1000
500
0
count
no
yes
housing


deposit
no
yes
3500
3000
2500
2000
1500
1000
500
0
count
no
yes
loan

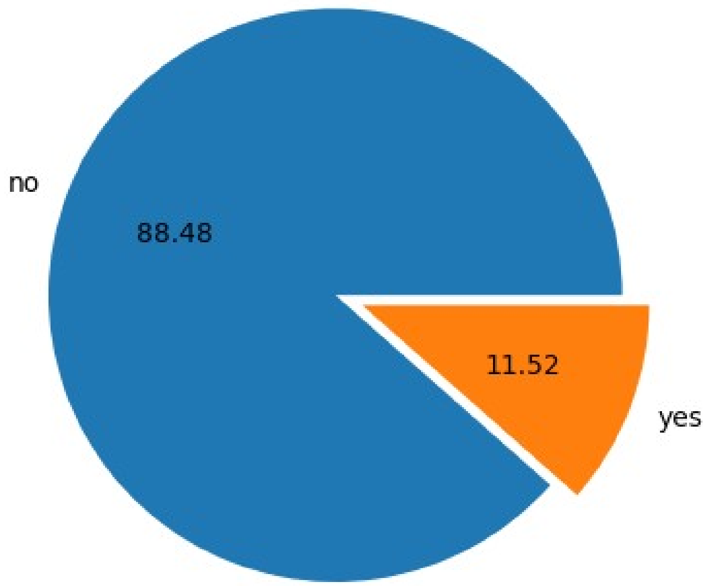

deposit
no
yes
count
2500
2000
1500
1000
500
0
cellular
unknown
telephone
contact


deposit
no
yes
count
1200
1000
800
600
400
200
0
oct
may
apr
jun
feb
aug
jan
jul
nov
sep
mar
dec
month



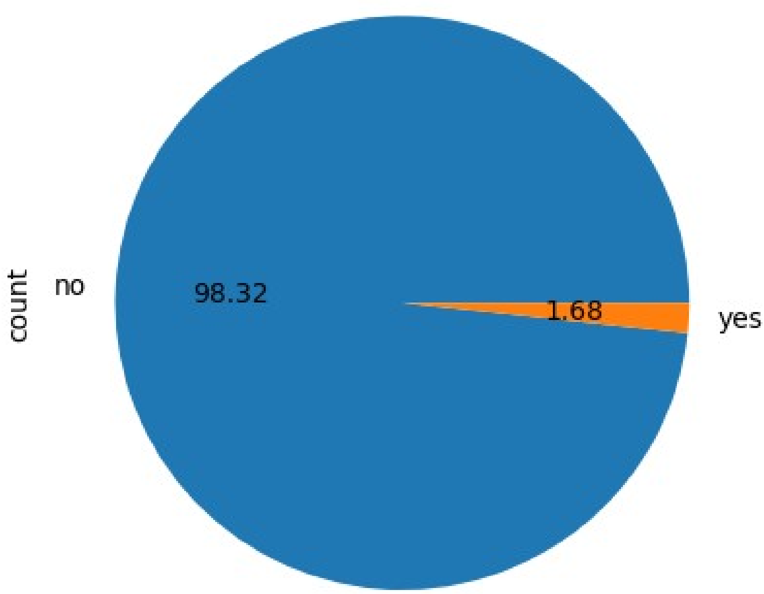



```
#Distribution of Outcome(Term Depositsts)
df['deposit'].value_counts()
keys= df['deposit'].value_counts().keys()
data= df['deposit'].value_counts().values
explode=[0,0.1]
plt.pie(data, labels=keys, explode=explode, autopct='%.2f')
plt.show()
```

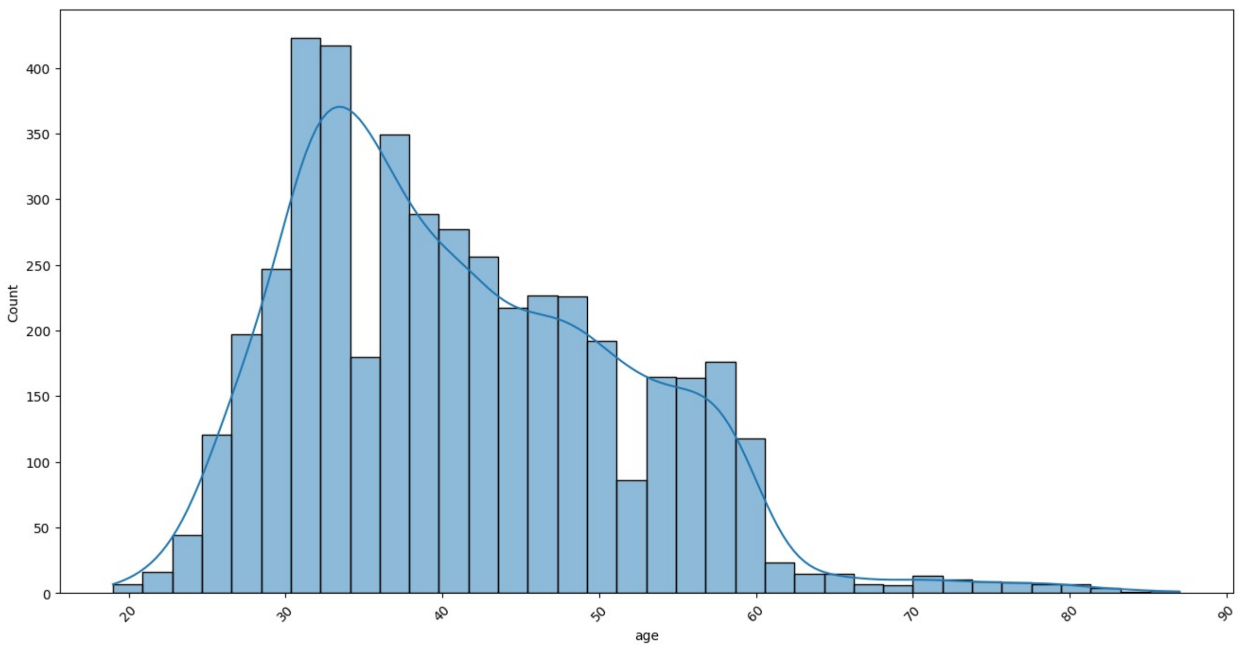


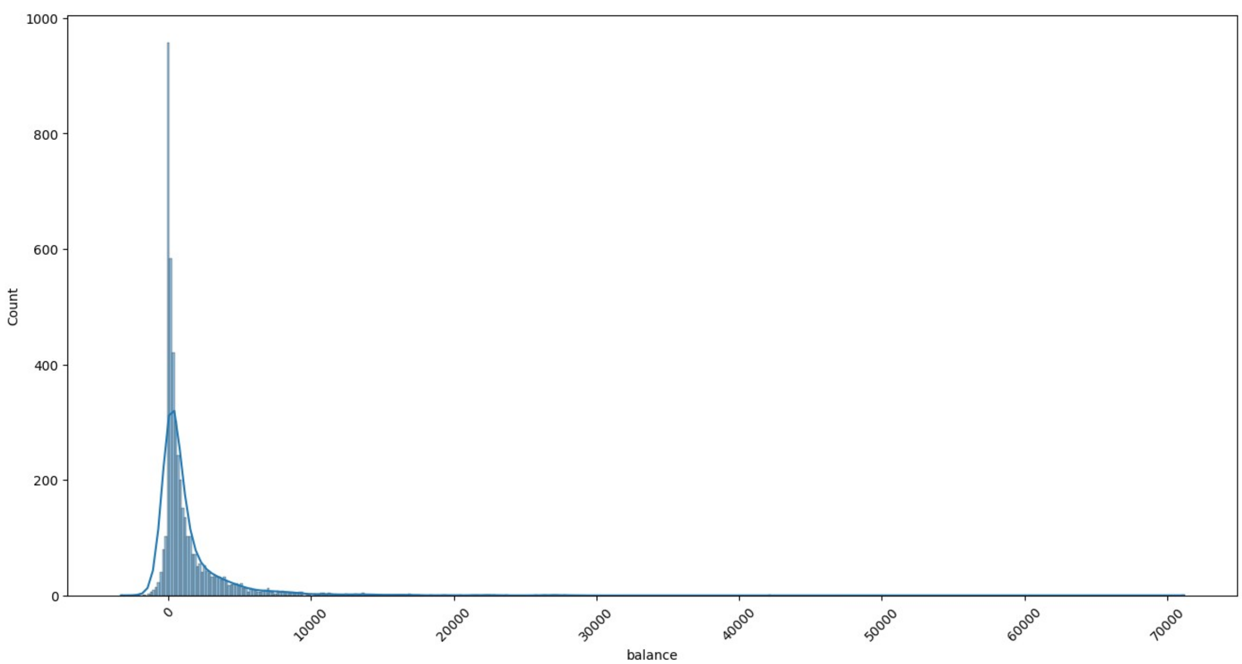

```
df['default'].value_counts().plot(kind='pie', autopct='%.2f')
```

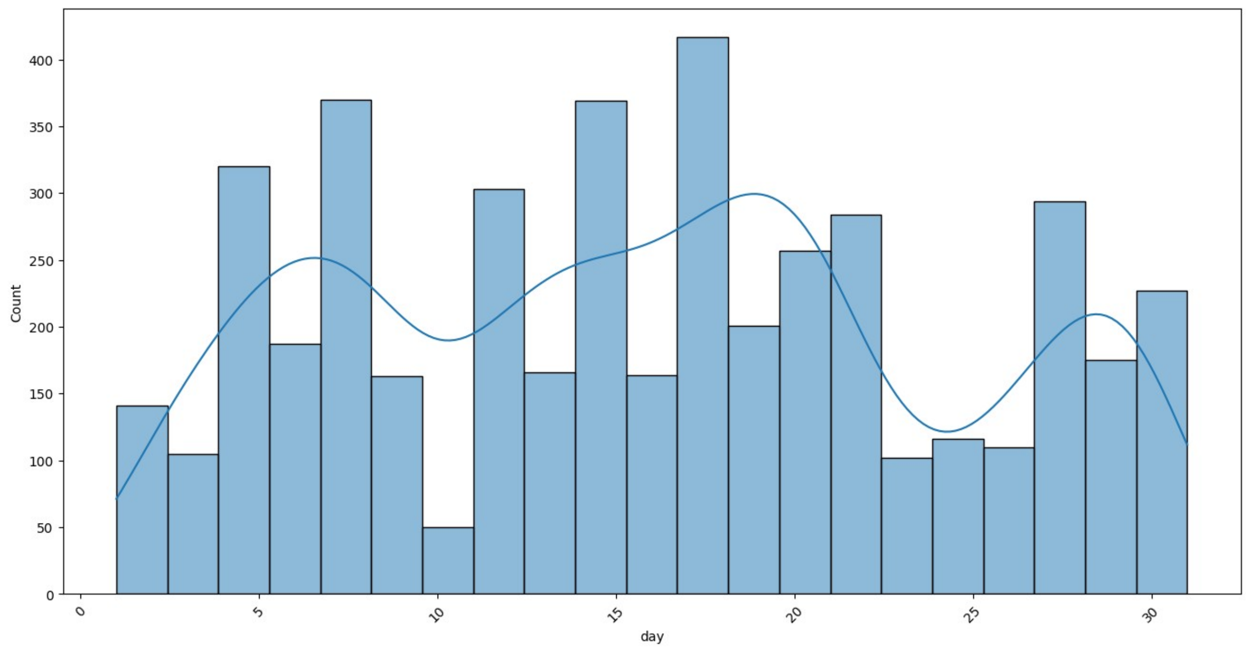
```
<Axes: ylabel='count'>
```



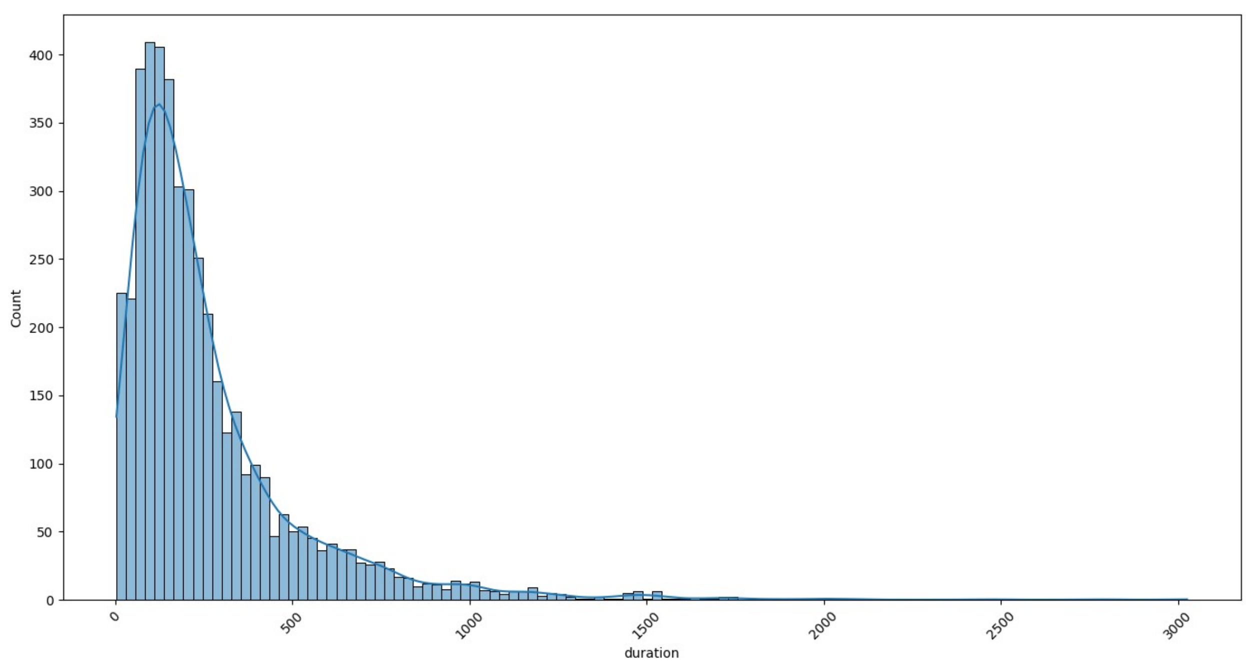
```python
num= [i for i in df.columns if df[i].dtypes=='int64' or
df[i].dtypes=='float']
for i, features in enumerate(num):
    plt.figure(figsize=(16,8))
    sns.histplot(x=features, kde=True, data=df)
    plt.xticks(rotation=45)
plt.show()
```



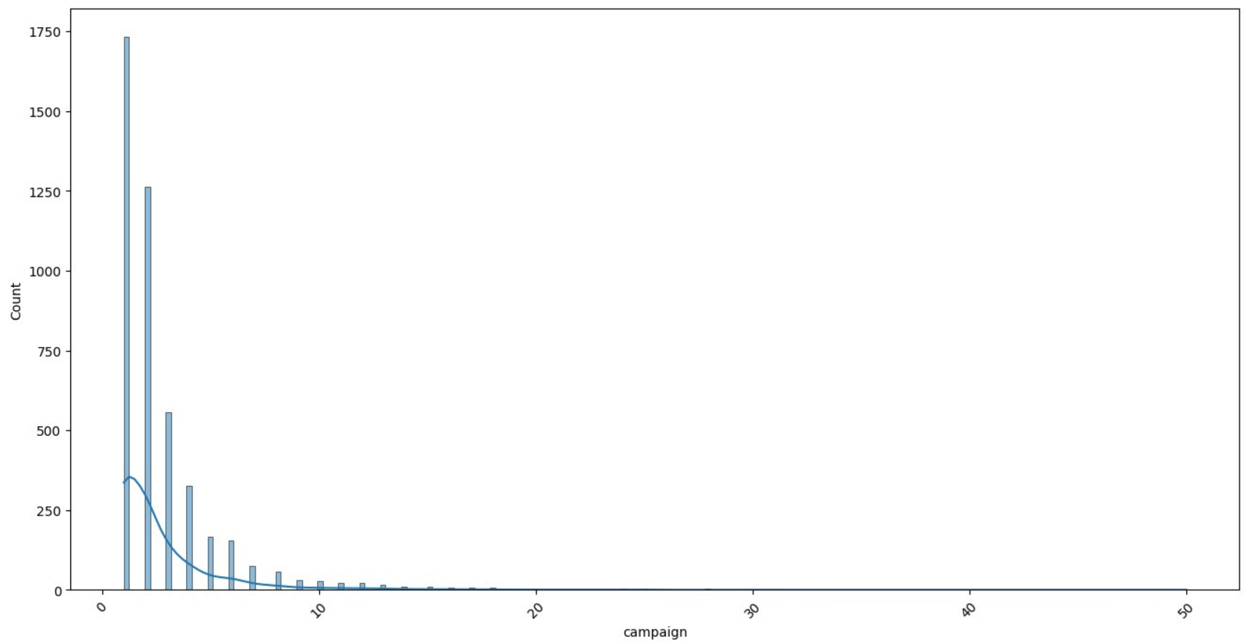




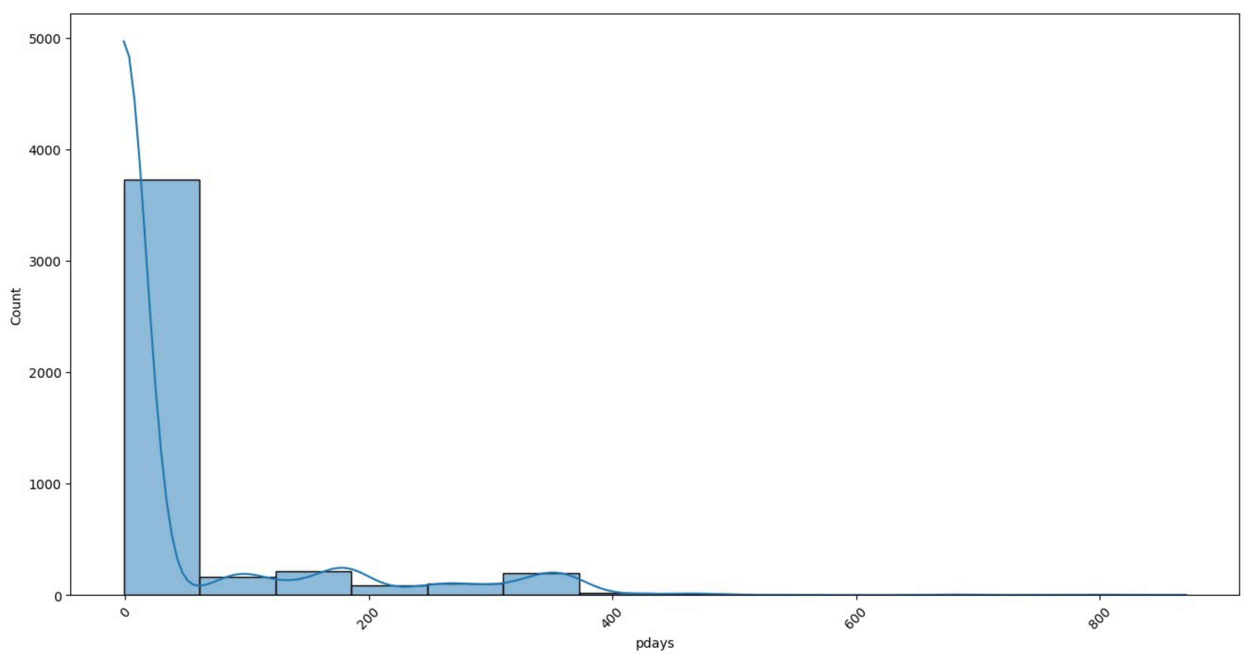
Count
day



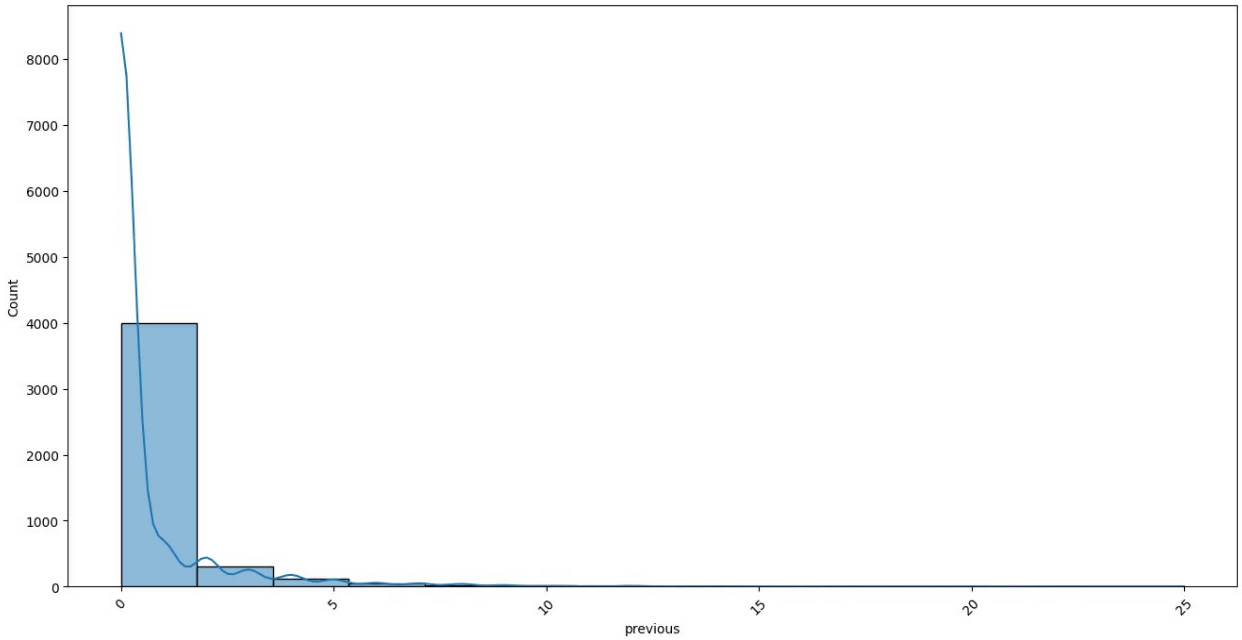
Count
duration



1750
1500
1250
1000
750
500
250
0
Count
0
10
20
30
40
50
campaign



5000
4000
3000
2000
1000
0
Count
0
200
400
600
800
pdays

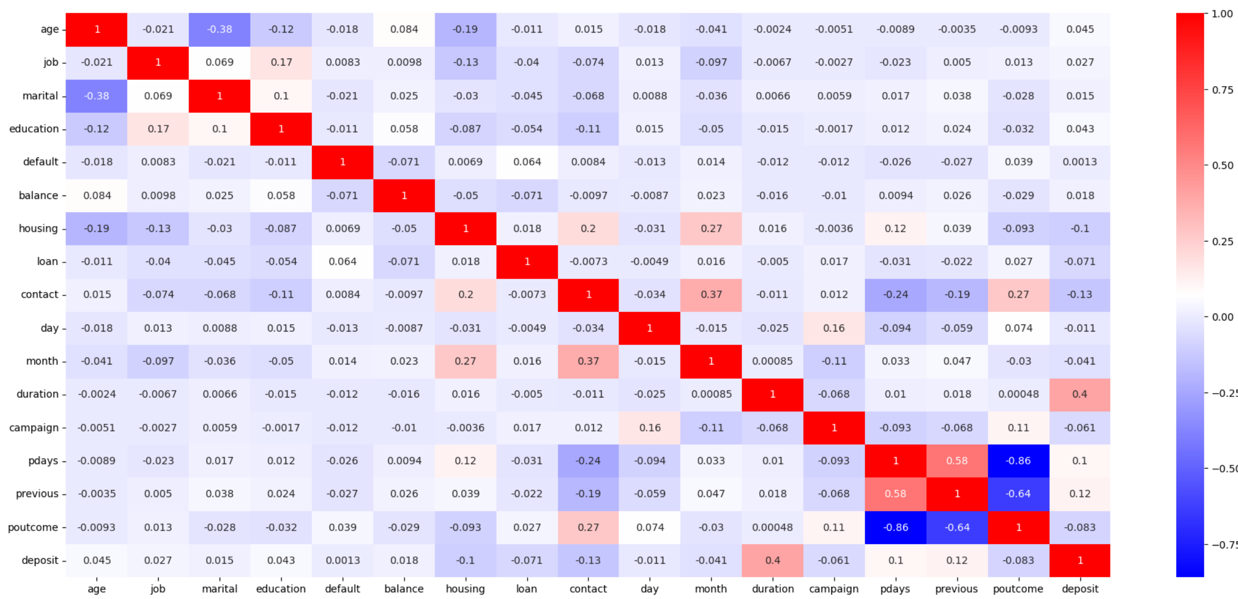



```
#Label encoding categorical features
le= LabelEncoder()
df[cat]= df[cat].apply(le.fit_transform)

df.head(3)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 4521,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 19,\n
\"max\": 87,\n        \"num_unique_values\": 67,\n        \"samples\":
[\n          50,\n          44,\n          36\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"job\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 3,\n        \"min\": 0,\n
\"max\": 11,\n        \"num_unique_values\": 12,\n        \"samples\":
[\n          5,\n          3,\n          10\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"marital\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 2,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1,\n          2,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"education\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          1,\n
3,\n          0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
```

```
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"balance\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 3009,\n        \"min\": -3313,\n        \"max\":
71188,\n        \"num_unique_values\": 2353,\n        \"samples\": [\n
1988,\n          7010\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"housing\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"loan\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"contact\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          0,\n
2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"day\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 8,\n        \"min\": 1,\n        \"max\": 31,\n
\"num_unique_values\": 31,\n        \"samples\": [\n          10,\n
18\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"month\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 3,\n        \"min\": 0,\n        \"max\": 11,\n
\"num_unique_values\": 12,\n        \"samples\": [\n          7,\n
11\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"duration\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 259,\n        \"min\": 4,\n
\"max\": 3025,\n        \"num_unique_values\": 875,\n
\"samples\": [\n          1516,\n          722\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"campaign\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3,\n
\"min\": 1,\n        \"max\": 50,\n        \"num_unique_values\": 32,\
n        \"samples\": [\n          28,\n          8\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"pdays\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 100,\n
\"min\": -1,\n        \"max\": 871,\n        \"num_unique_values\":
292,\n        \"samples\": [\n          63,\n          385\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"previous\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1,\n        \"min\": 0,\n
\"max\": 25,\n        \"num_unique_values\": 24,\n        \"samples\":
[\n          6,\n          11\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"poutcome\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 3,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          0,\n          2\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"deposit\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}

#Correlation Analysis using Heatmap
plt.figure(figsize=(23,10))
sns.heatmap(df.corr(), annot=True, cmap= 'bwr')
plt.show()
```




```
#Standarisation
x= df.drop("deposit", axis =1)
y= df['deposit']

scaler=  StandardScaler()

X_Scaled= pd.DataFrame(scaler.fit_transform(x), columns=x.columns)

X_Scaled.head(2)
```

{"summary":"{\n \"name\": \"X_Scaled\",\n \"rows\": 4521,\n
\"fields\": [\n   {\n   \"column\": \"age\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.0001106133513697,\n   \"min\": -2.0964547234147863,\n
\"max\": 4.333780260871122,\n   \"num_unique_values\": 67,\n
\"samples\": [\n   0.8349759311861424,\n
0.26760225610209176,\n   -0.48889597734330914\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"job\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 1.00011061335137,\n
\"min\": -1.355056673996865,\n   \"max\": 2.0239897811271437,\n
\"num_unique_values\": 12,\n   \"samples\": [\n
0.1808735328776843,\n   -0.4334985498721355,\n
1.7168037397522335\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"marital\",\n   \"properties\": {\n   \"dtype\": \"number\",\
n   \"std\": 1.00011061335137,\n   \"min\": -
1.9142545494396246,\n   \"max\": 1.4213957947563831,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   -
0.24642937734162074,\n   1.4213957947563831,\n   -
1.9142545494396246\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"education\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 1.00011061335137,\n   \"min\": -
1.6447553471316043,\n   \"max\": 2.362396938326623,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   -
0.3090379186455286,\n   2.362396938326623,\n   -
1.6447553471316043\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"default\",\n   \"properties\": {\n   \"dtype\": \"number\",\
n   \"std\": 1.0001106133513695,\n   \"min\": -
0.13075879613683394,\n   \"max\": 7.647669063529301,\n
\"num_unique_values\": 2,\n   \"samples\": [\n
7.647669063529301,\n   -0.13075879613683394\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"balance\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 1.00011061335137,\
n   \"min\": -1.573671458777467,\n   \"max\":
23.18320537433136,\n   \"num_unique_values\": 2353,\n
\"samples\": [\n   0.18786468291606623,\n
1.856688396099413\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"housing\",\n   \"properties\": {\n   \"dtype\": \"number\",\
n   \"std\": 1.00011061335137,\n   \"min\": -
1.1420513760622808,\n   \"max\": 0.8756173504627571,\n
\"num_unique_values\": 2,\n   \"samples\": [\n
0.8756173504627571,\n   -1.1420513760622808\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"loan\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 1.0001106133513697,\n

```
\"min\": -0.42475611186796425,\n        \"max\": 2.354292197473666,\n
\"num_unique_values\": 2,\n        \"samples\": [\n
2.354292197473666,\n          -0.42475611186796425\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"contact\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
1.0001106133513697,\n        \"min\": -0.7236415227048494,\n
\"max\": 1.4951331969619017,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          -0.7236415227048494,\n
1.4951331969619017\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"day\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1.0001106133513697,\n        \"min\": -1.8086246046175192,\n
\"max\": 1.82916984199943,\n        \"num_unique_values\": 31,\n
\"samples\": [\n          -0.7172862706324343,\n
0.25279224846541876\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"month\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1.0001106133513697,\n        \"min\": -1.84522023273306 2,\n
\"max\": 1.8184779105195394,\n        \"num_unique_values\": 12,\n
\"samples\": [\n          0.4862240402458662,\n
1.8184779105195394\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"duration\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001106133513697,\n        \"min\": -
1.0005134150831179,\n        \"max\": 10.626413834829654,\n
\"num_unique_values\": 875,\n        \"samples\": [\n
4.818723262132411,\n          1.7628542662930393\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"campaign\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
1.0001106133513697,\n        \"min\": -0.5768294699140059,\n
\"max\": 15.18151997398534,\n        \"num_unique_values\": 32,\n
\"samples\": [\n          8.106342672642777,\n
1.6743633077859008\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"pdays\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1.0001106133513697,\n        \"min\": -0.4072182979332062,\n
\"max\": 8.30319580991816,\n        \"num_unique_values\": 292,\n
\"samples\": [\n          0.23207815034946291,\n
3.4485384057716417\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"previous\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001106133513697,\n        \"min\": -
0.3204128219555523,\n        \"max\": 14.44300307756597,\n
\"num_unique_values\": 24,\n        \"samples\": [\n
3.2228069939296136,\n          6.175490173833919\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"poutcome\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
```

1.0001106133513697,\n   \"min\": -2.5799607112523955,\n
\"max\": 0.44441328414226683,\n   \"num_unique_values\": 4,\n
\"samples\": [\n   -2.5799607112523955,\n   -
0.5637113809892873\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"X_Scaled"}

```
#Model building - decision Tree Classifier
#Train_Test_Split
x_train, x_test, y_train, y_test= train_test_split(X_Scaled, y,
test_size=0.3)


dt= DecisionTreeClassifier()
dt.fit(x_train, y_train)
```

```
DecisionTreeClassifier()
```

```
print('Train Score: {}'.format(dt.score(x_train, y_train)))
print('Test Score: {}'.format(dt.score(x_test, y_test)))
```

```
Train Score: 1.0
Test Score: 0.8585114222549742
```

```
cross_val_score(dt, x_train, y_train, cv=5).mean()
```

```
0.8666181734557161
```

```
ypred= dt.predict(x_test)
ypred
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
#Hyperparameter tunning
#Applying Grid search cv to find best estimaters to improve model
performance
param_grid= {
    'criterion': ['gini', 'entropy'],
    'max_depth': [3, 5, 7,10, None],
    'min_samples_leaf': [3,5,7,9,10,20]
    }

gscv= GridSearchCV(dt, param_grid, cv=5, verbose=1)
gscv.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 60 candidates, totalling 300 fits
```

```
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [3, 5, 7, 10, None],
                         'min_samples_leaf': [3, 5, 7, 9, 10, 20]},
             verbose=1)
```

```
gscv.best_params_
```

```
{'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 3}
```

```
gscv.best_estimator_
```

```
DecisionTreeClassifier(max_depth=5, min_samples_leaf=3)
```

```
cross_val_score(gscv.best_estimator_, x_train, y_train, cv=5).mean()
```

```
0.8947502349671046
```

```
clf = DecisionTreeClassifier(criterion='gini', max_depth=5,
min_samples_leaf=3)
clf.fit(x_train, y_train)
```

```
DecisionTreeClassifier(max_depth=5, min_samples_leaf=3)
```

```
y_pred= clf.predict(x_test)
y_pred
```

```
array([1, 0, 0, ..., 0, 0, 1])
```

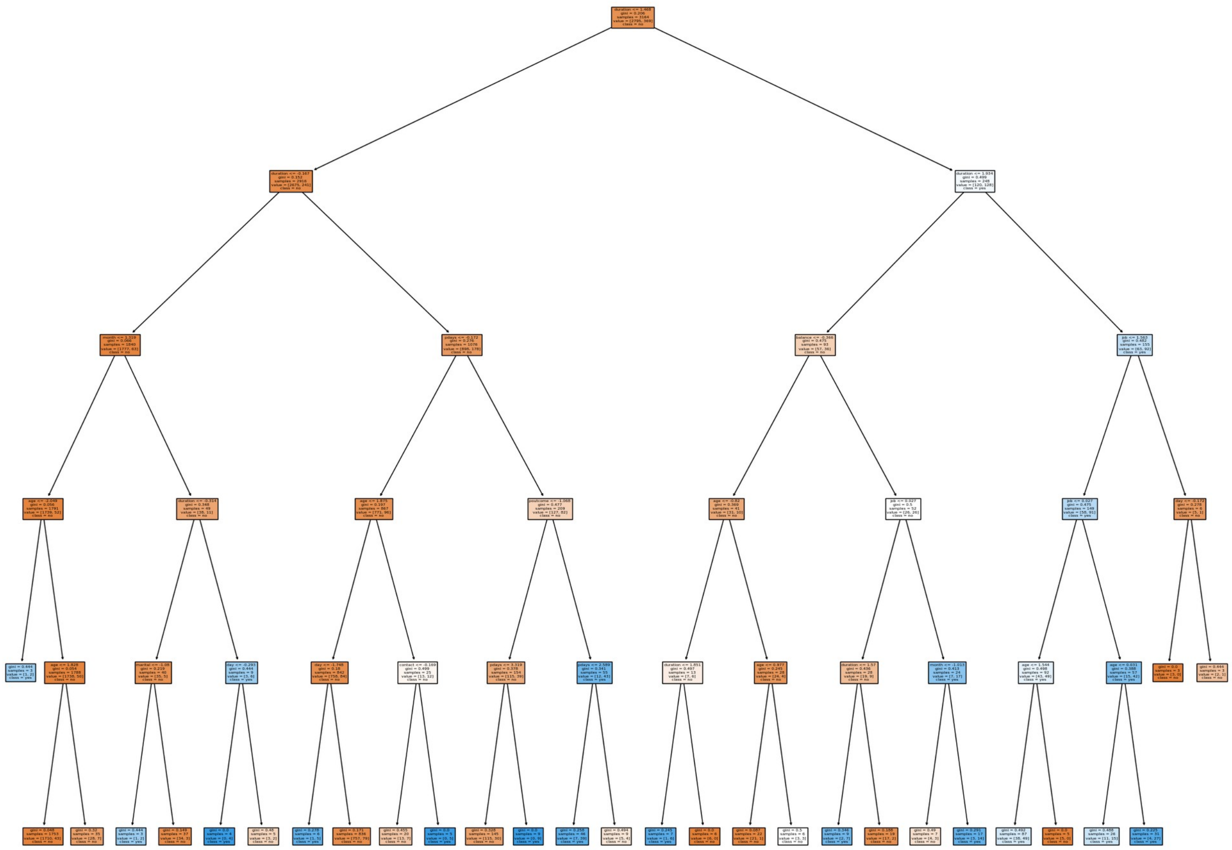
```
#Accuracy Score
accuracy = accuracy_score(y_test, y_pred)
print("Test Accuracy of Decision Tree Classifier:
{}".format(accuracy*100))
```

```
Test Accuracy of Decision Tree Classifier: 88.94620486366986
```

```
#Cross Validation Score
Cross_val= cross_val_score(clf, x_train, y_train, cv=5).mean()
print("Cross Validation Score of Decision Tree Classifier:
",Cross_val*100)
```

```
Cross Validation Score of Decision Tree Classifier:  89.4434279200912
```

```
#Visualize the decision tree
from sklearn import tree
fig= plt.figure(figsize=(25,20))
tree.plot_tree(clf, filled=True,
feature_names=x_train.columns,class_names=['no', 'yes'])
plt.show()
```

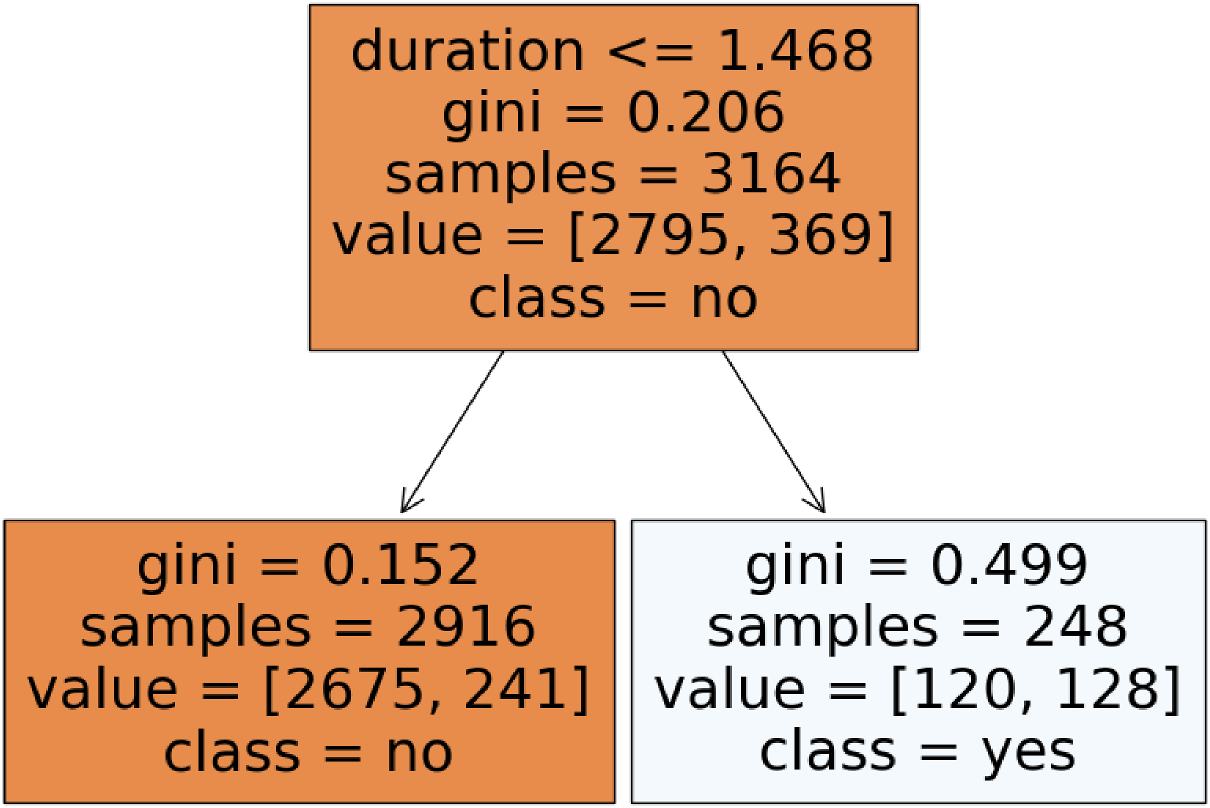


```
#create a classifier with pruning enabled
dt= DecisionTreeClassifier(ccp_alpha=0.01)
dt.fit(x_train, y_train)
```

```
DecisionTreeClassifier(ccp_alpha=0.01)
```

```
#make the classifier
ypred= dt.predict(x_test)

#calculate accuracy
accuracy= accuracy_score(y_test, ypred)
print("Accuracy:", accuracy)
#visualize the pruned decision tree
plt.figure(figsize=(10,8))
tree.plot_tree(dt, filled=True, feature_names=x_train.columns,
class_names=['no', 'yes'])
plt.show()
```

```
Accuracy: 0.8828297715549005
```


duration <= 1.468
gini = 0.206
samples = 3164
value = [2795, 369]
class = no
gini = 0.152
samples = 2916
value = [2675, 241]
class = no
gini = 0.499
samples = 248
value = [120, 128]
class = yes