In [4]:
```
pip install pandas
```

```
Collecting pandas
  Downloading pandas-2.1.1-cp310-cp310-win_amd64.whl (10.7 MB)
     ---------------------------------------- 10.7/10.7 MB 4.5 MB/s eta 0:00:00
Requirement already satisfied: pytz>=2020.1 in c:\users\ram\appdata\local\progr
ams\python\python310\lib\site-packages (from pandas) (2022.2.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\ram\appdata\l
ocal\programs\python\python310\lib\site-packages (from pandas) (2.8.2)
Collecting numpy>=1.22.4
  Downloading numpy-1.26.0-cp310-cp310-win_amd64.whl (15.8 MB)
     ---------------------------------------- 15.8/15.8 MB 5.2 MB/s eta 0:00:00
Collecting tzdata>=2022.1
  Downloading tzdata-2023.3-py2.py3-none-any.whl (341 kB)
     ---------------------------------------- 341.8/341.8 KB 4.3 MB/s eta 0:00:00
Requirement already satisfied: six>=1.5 in c:\users\ram\appdata\local\programs
\python\python310\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.1
6.0)
Installing collected packages: tzdata, numpy, pandas
Successfully installed numpy-1.26.0 pandas-2.1.1 tzdata-2023.3
Note: you may need to restart the kernel to use updated packages.

WARNING: You are using pip version 22.0.4; however, version 23.2.1 is availabl
e.
You should consider upgrading via the 'C:\Users\RAM\AppData\Local\Programs\Pyth
on\Python310\python.exe -m pip install --upgrade pip' command.
```

In [5]:
```
pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.0-py3-none-any.whl (294 kB)
     ---------------------------------------- 294.6/294.6 KB 2.0 MB/s eta 0:00:00
```

In [5]:
```
pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.0-py3-none-any.whl (294 kB)
     ---------------------------------------- 294.6/294.6 KB 2.0 MB/s eta 0:00:00
Collecting matplotlib!=3.6.1,>=3.3
  Downloading matplotlib-3.8.0-cp310-cp310-win_amd64.whl (7.6 MB)
     ---------------------------------------- 7.6/7.6 MB 4.7 MB/s eta 0:00:00
Requirement already satisfied: pandas>=1.2 in c:\users\ram\appdata\local\progra
ms\python\python310\lib\site-packages (from seaborn) (2.1.1)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\ram\appdata\loc
al\programs\python\python310\lib\site-packages (from seaborn) (1.26.0)
Requirement already satisfied: packaging>=20.0 in c:\users\ram\appdata\local\pr
ograms\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seabor
n) (21.3)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\ram\appdata\local\p
rograms\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seabo
rn) (3.0.9)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.4.5-cp310-cp310-win_amd64.whl (56 kB)
     ---------------------------------------- 56.1/56.1 KB 975.6 kB/s eta 0:00:00
Collecting fonttools>=4.22.0
  Downloading fonttools-4.43.1-cp310-cp310-win_amd64.whl (2.1 MB)
     ---------------------------------------- 2.1/2.1 MB 2.1 MB/s eta 0:00:00
Collecting contourpy>=1.0.1
  Downloading contourpy-1.1.1-cp310-cp310-win_amd64.whl (477 kB)
     ---------------------------------------- 478.0/478.0 KB 1.7 MB/s eta 0:00:00
Collecting cycler>=0.10
  Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Collecting pillow>=6.2.0
  Downloading Pillow-10.0.1-cp310-cp310-win_amd64.whl (2.5 MB)
     ---------------------------------------- 2.5/2.5 MB 2.9 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.7 in c:\users\ram\appdata\loc
al\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.3->s
eaborn) (2.8.2)
Requirement already satisfied: tzdata>=2022.1 in c:\users\ram\appdata\local\pro
grams\python\python310\lib\site-packages (from pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\ram\appdata\local\progr
ams\python\python310\lib\site-packages (from pandas>=1.2->seaborn) (2022.2.1)
Requirement already satisfied: six>=1.5 in c:\users\ram\appdata\local\programs
\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.
6.1,>=3.3->seaborn) (1.16.0)
Installing collected packages: pillow, kiwisolver, fonttools, cycler, contourp
y, matplotlib, seaborn
Successfully installed contourpy-1.1.1 cycler-0.12.1 fonttools-4.43.1 kiwisolve
r-1.4.5 matplotlib-3.8.0 pillow-10.0.1 seaborn-0.13.0
Note: you may need to restart the kernel to use updated packages.

WARNING: You are using pip version 22.0.4; however, version 23.2.1 is availabl
e.
You should consider upgrading via the 'C:\Users\RAM\AppData\Local\Programs\Pyth
on\Python310\python.exe -m pip install --upgrade pip' command.
```

In [6]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\ram\appdata\local\programs\pyt
hon\python310\lib\site-packages (1.26.0)
```

In [6]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\ram\appdata\local\programs\pyt
hon\python310\lib\site-packages (1.26.0)

WARNING: You are using pip version 22.0.4; however, version 23.2.1 is availabl
e.
You should consider upgrading via the 'C:\Users\RAM\AppData\Local\Programs\Pyth
on\Python310\python.exe -m pip install --upgrade pip' command.
```

In [10]:
```
pip install -U scikit-learn
```

```
Collecting scikit-learn
  Downloading scikit_learn-1.3.1-cp310-cp310-win_amd64.whl (9.3 MB)
     ---------------------------------------- 9.3/9.3 MB 4.0 MB/s eta 0:00:00
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-3.2.0-py3-none-any.whl (15 kB)
Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\users\ram\appdata\local
\programs\python\python310\lib\site-packages (from scikit-learn) (1.26.0)
Collecting joblib>=1.1.1
  Downloading joblib-1.3.2-py3-none-any.whl (302 kB)
     ---------------------------------------- 302.2/302.2 KB 2.7 MB/s eta 0:00:00
Collecting scipy>=1.5.0
  Downloading scipy-1.11.3-cp310-cp310-win_amd64.whl (44.1 MB)
     ---------------------------------------- 44.1/44.1 MB 3.0 MB/s eta 0:00:00
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
Successfully installed joblib-1.3.2 scikit-learn-1.3.1 scipy-1.11.3 threadpoolc
tl-3.2.0
Note: you may need to restart the kernel to use updated packages.

WARNING: You are using pip version 22.0.4; however, version 23.2.1 is availabl
e.
You should consider upgrading via the 'C:\Users\RAM\AppData\Local\Programs\Pyth
on\Python310\python.exe -m pip install --upgrade pip' command.
```

In [23]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
import csv
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
%matplotlib inline
```

In [30]:
```
df = pd.read_csv(r'G:\programming\bank-additional\bank-additional.csv',delimiter=
df.rename (columns={'y':'deposit'}, inplace=True)
```

In [31]:
```
df.head()
```

Out[31]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_wee

In [31]:
```
df.head()
```

Out[31]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_wee
0	30	blue-collar	married	basic.9y	no	yes	no	cellular	may	f
1	39	services	single	high.school	no	no	no	telephone	may	f
2	25	services	married	high.school	no	yes	no	telephone	jun	we
3	38	services	married	basic.9y	no	unknown	unknown	telephone	jun	f
4	47	admin.	married	university.degree	no	yes	no	cellular	nov	mo

5 rows × 21 columns

In [32]:
```
df.tail()
```

Out[32]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week
4114	30	admin.	married	basic.6y	no	yes	yes	cellular	jul	thu
4115	39	admin.	married	high.school	no	yes	no	telephone	jul	fri
4116	27	student	single	high.school	no	no	no	cellular	may	mon
4117	58	admin.	married	high.school	no	no	no	cellular	aug	fri
4118	34	management	single	high.school	no	yes	no	cellular	nov	wed

5 rows × 21 columns

In [33]:
```
df.shape
```

Out[33]:
```
(4119, 21)
```

In [34]:
```
df.columns
```

Out[34]:
```
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')
```

In [37]:
```
df.dtypes.value_counts()
```

Out[37]:
```
object     11
int64       5
float64     5
Name: count, dtype: int64
```

In [38]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4119 entries, 0 to 4118
Data columns (total 21 columns):
```

In [38]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4119 entries, 0 to 4118
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             4119 non-null   int64  
 1   job             4119 non-null   object 
 2   marital         4119 non-null   object 
 3   education       4119 non-null   object 
 4   default         4119 non-null   object 
 5   housing         4119 non-null   object 
 6   loan            4119 non-null   object 
 7   contact         4119 non-null   object 
 8   month           4119 non-null   object 
 9   day_of_week     4119 non-null   object 
 10  duration        4119 non-null   int64  
 11  campaign        4119 non-null   int64  
 12  pdays           4119 non-null   int64  
 13  previous        4119 non-null   int64  
 14  poutcome        4119 non-null   object 
 15  emp.var.rate    4119 non-null   float64
 16  cons.price.idx  4119 non-null   float64
 17  cons.conf.idx   4119 non-null   float64
 18  euribor3m       4119 non-null   float64
 19  nr.employed     4119 non-null   float64
 20  deposit         4119 non-null   object 
dtypes: float64(5), int64(5), object(11)
memory usage: 675.9+ KB
```

In [39]:
```
df.duplicated().sum()
```

Out[39]:
```
0
```

In [40]:
```
df.isna().sum()
```

Out[40]:
```
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
```

In [42]:
```
cat_cols = df.select_dtypes(include='object').columns
print(cat_cols)
num_cols = df.select_dtypes(exclude='object').columns
print(num_cols)
```

deposit 0
dtype: int64

In [42]:
```
cat_cols = df.select_dtypes(include='object').columns
print(cat_cols)
num_cols = df.select_dtypes(exclude='object').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')
```

In [43]:
```
df.describe()
```

Out[43]:

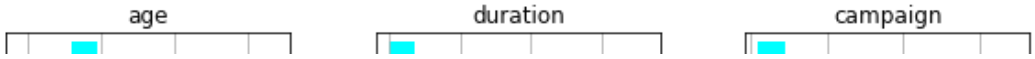
	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx
count	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000
mean	40.113620	256.788055	2.537266	960.422190	0.190337	0.084972	93.579704
std	10.313362	254.703736	2.568159	191.922786	0.541788	1.563114	0.579349
min	18.000000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000
25%	32.000000	103.000000	1.000000	999.000000	0.000000	-1.800000	93.075000
50%	38.000000	181.000000	2.000000	999.000000	0.000000	1.100000	93.749000
75%	47.000000	317.000000	3.000000	999.000000	0.000000	1.400000	93.994000
max	88.000000	3643.000000	35.000000	999.000000	6.000000	1.400000	94.767000

In [44]:
```
df.describe(include= 'object')
```

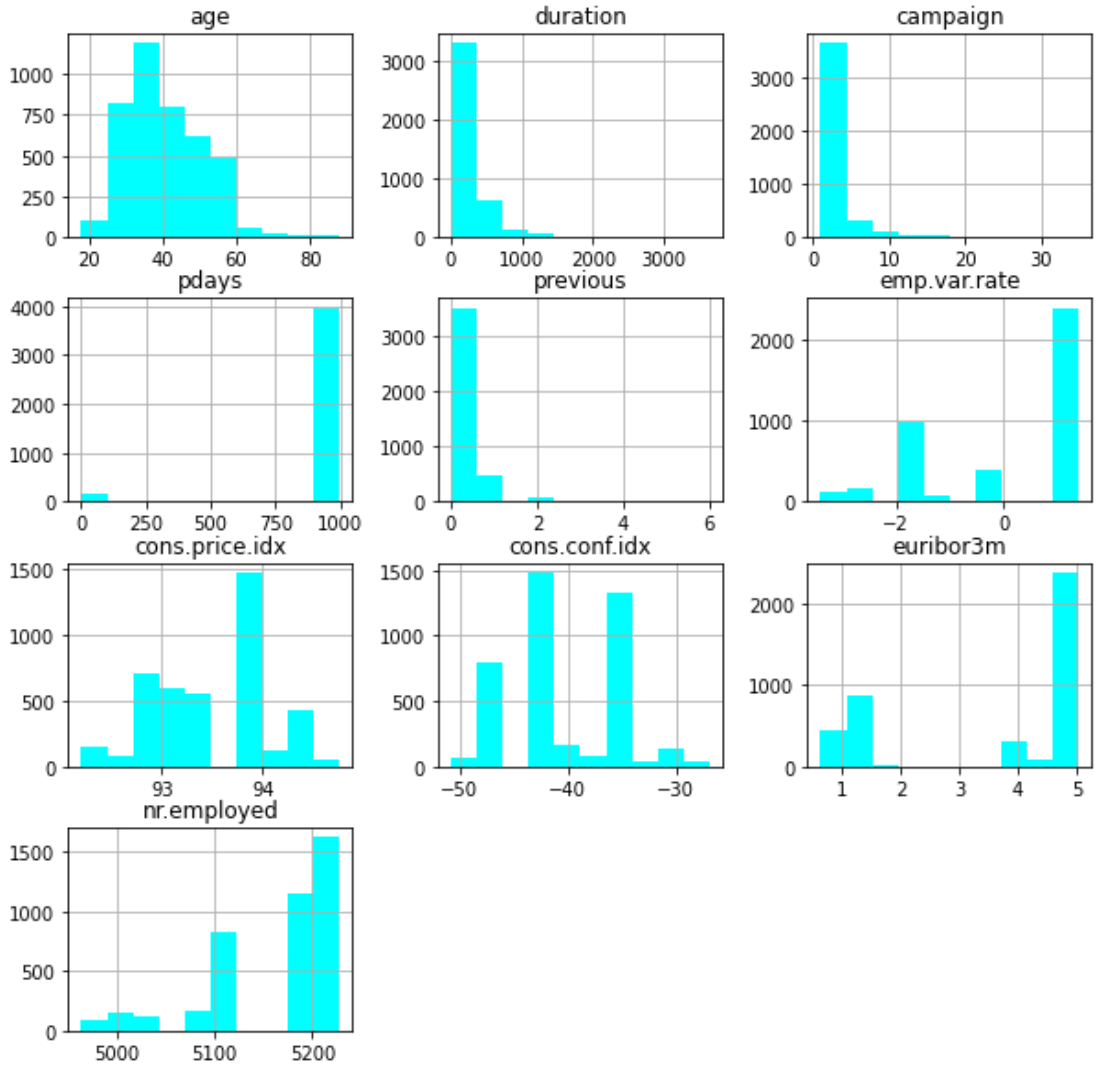
Out[44]:

	job	marital	education	default	housing	loan	contact	month	day_of_week	po
count	4119	4119	4119	4119	4119	4119	4119	4119	4119	
unique	12	4	8	3	3	3	2	10	5	
top	admin.	married	university.degree	no	yes	no	cellular	may	thu	non
freq	1012	2509	1264	3315	2175	3349	2652	1378	860	

In [45]:
```
df.hist(figsize=(10, 10), color="#00FFFF")
plt.show()
```

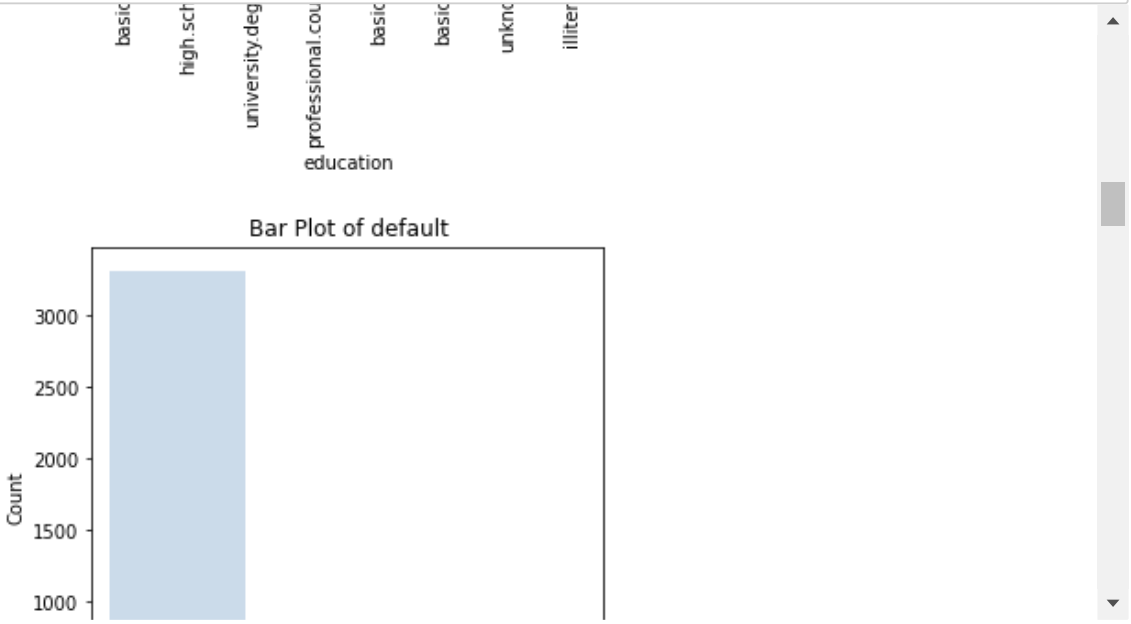


In [45]:
```
df.hist(figsize=(10, 10), color="#00FFFF")
plt.show()
```



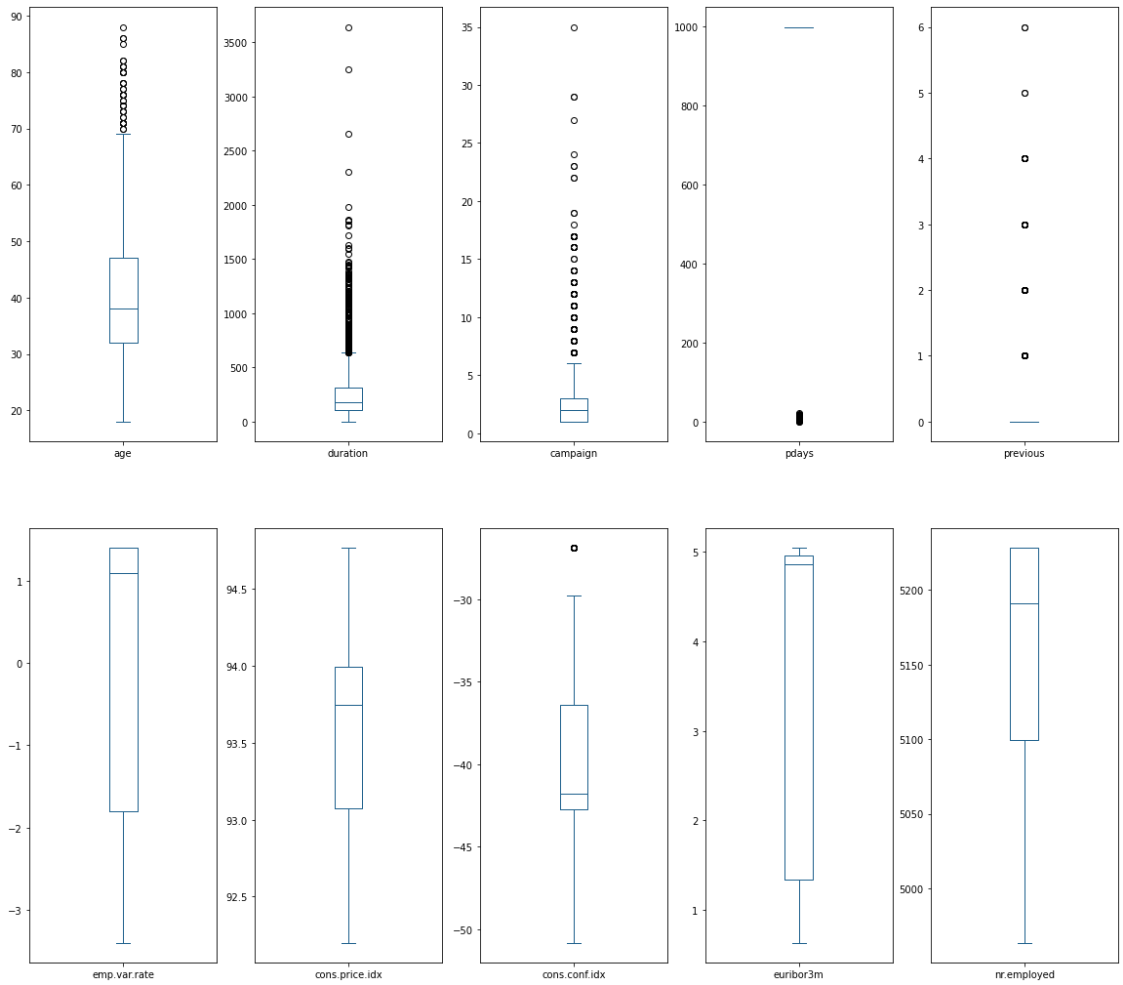
In [53]:
```
for feature in cat_cols:
      plt.figure(figsize=(5, 5))
      sns.countplot(x=feature, data=df, palette='Blues')
      plt.title(f'Bar Plot of {feature}')
      plt.xlabel(feature)
```

In [53]:
```
for feature in cat_cols:
      plt.figure(figsize=(5, 5))
      sns.countplot(x=feature, data=df, palette='Blues')
      plt.title(f'Bar Plot of {feature}')
      plt.xlabel(feature)
      plt.ylabel('Count')
      plt.xticks(rotation=90)
      plt.show()
```



In [54]:
```
df.plot(kind="box", subplots=True, layout=(2, 5), figsize=(20, 18), color="#1F618
plt.show()
```

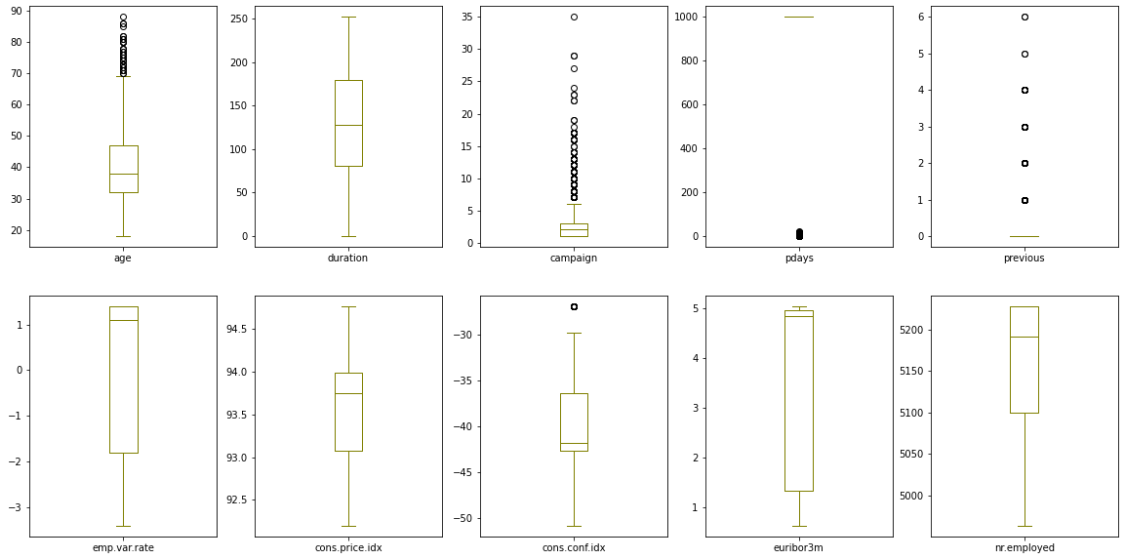
In [54]:
```
df.plot(kind="box", subplots=True, layout=(2, 5), figsize=(20, 18), color="#1F618
plt.show()
```



In [62]:
```
column = df[['age','campaign', 'duration']]
q1 = np.percentile (column, 25)
q3 = np.percentile (column, 75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
df[['age','campaign','duration']] = column[(column> lower_bound) & (column < uppe
```

In [65]:
```
df.plot(kind='box', subplots=True, layout=(2,5), figsize=(20,10), color= '#808000
plt.show()
```

In [65]:
```
df.plot(kind='box', subplots=True, layout=(2,5), figsize=(20,10), color= '#808000
plt.show()
```



In [68]:
```
corr = df.corr()
print(corr)
corr =corr[abs(corr) >= 0.90]
sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=0.2)
plt.show()
```

In [68]:
```
corr = df.corr()
print(corr)
corr =corr[abs(corr) >= 0.90]
sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=0.2)
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [68], in <cell line: 1>()
----> 1 corr = df.corr()
      2 print(corr)
      3 corr =corr[abs(corr) >= 0.90]

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py:10707, in DataFrame.corr(self, method, min_periods, numeric_only)
  10705 cols = data.columns
  10706 idx = cols.copy()
> 10707 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  10709 if method == "pearson":
  10710     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py:1892, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1890 if dtype is not None:
   1891     dtype = np.dtype(dtype)
-> 1892 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1893 if result.dtype is not dtype:
   1894     result = np.array(result, dtype=dtype, copy=False)

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\internals\managers.py:1656, in BlockManager.as_array(self, dtype, copy, na_value)
   1654         arr.flags.writeable = False
   1655 else:
-> 1656     arr = self._interleave(dtype=dtype, na_value=na_value)
   1657     # The underlying data was copied within _interleave, so no need
   1658     # to further copy if copy=True or setting na_value
   1660 if na_value is lib.no_default:

File ~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\internals\managers.py:1715, in BlockManager._interleave(self, dtype, na_value)
   1713     else:
   1714         arr = blk.get_values(dtype)
-> 1715     result[rl.indexer] = arr
   1716     itemmask[rl.indexer] = 1
   1718 if not itemmask.all():

ValueError: could not convert string to float: 'blue-collar'
```

In [70]:
```
high_corr_cols = ['emp.var.rate', 'euribor3n', 'nr.employed']
```

In [71]:
```
df1 = df.copy()
df1.columns
```

Out[71]:
```
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')
```

In [74]:
```
df1.shape
```

Out[74]:
```
(4119, 21)
```

In [74]:
```
        'cons.conf.idx', 'euribor3m', 'nr.employed', 'deposit'],
      dtype='object')
df1.shape
```

Out[74]: (4119, 21)

In [76]:
```
from sklearn.preprocessing import LabelEncoder
lb= LabelEncoder()
df_encoded = df1.apply(lb.fit_transform)
df_encoded
```

Out[76]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campa
0	12	1	1	2	0	2	0	0	6	0	...	
1	21	7	2	3	0	0	0	1	6	0	...	
2	7	7	1	3	0	2	0	1	4	4	...	
3	20	7	1	2	0	1	1	1	4	0	...	
4	29	0	1	6	0	2	0	0	7	1	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4114	12	0	1	1	0	2	2	0	3	2	...	
4115	21	0	1	3	0	2	0	1	3	0	...	
4116	9	8	2	3	0	0	0	0	6	1	...	
4117	40	0	1	3	0	0	0	0	1	0	...	
4118	16	4	2	3	0	2	0	0	7	4	...	

4119 rows × 21 columns

In [77]:
```
df_encoded['deposit'].value_counts()
```

Out[77]:
```
deposit
0    3668
1     451
Name: count, dtype: int64
```

In [78]:
```
x = df_encoded.drop('deposit',axis=1)
y = df_encoded ['deposit']
print(x.shape)
print (y.shape)
print(type(x))
print (type(y))
```

```
(4119, 20)
(4119,)
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.series.Series'>
```

In [79]:
```
from sklearn.model_selection import train_test_split
```

In [80]:
```
print(4119*0.25)
```

```
1029.75
```

In [84]:
```
x_train,x_test,y_train,y_test =  train_test_split(x,y,test_size=0.25, random_stat
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

In [84]:
```
x_train,x_test,y_train,y_test =  train_test_split(x,y,test_size=0.25, random_stat
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(3089, 20)
(1030, 20)
(3089,)
(1030,)
```

In [85]:
```
from sklearn.metrics import confusion_matrix, classification_report,accuracy_scor
```

In [89]:
```
def eval_model(y_test,y_pred):
      acc = accuracy_score(y_test,y_pred)
      print('Accuracy Score', acc)
      cm = confusion_matrix(y_test,y_pred)
      print('Confusion Matrix\n', cn)
      print('Classification Report\n', classification_report(y_test,y_pred))

def escore(model):
      train_score = model.score(x_train,y_train)
      test_score = model.score(x_test,y_test)
      print('Training Score', train_score)
      print('Testing Score', test_score)
```

In [92]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [93]:
```
dt = DecisionTreeClassifier(criterion='gini', max_depth=5,min_samples_split=10)
dt.fit(x_train,y_train)
```

Out[93]:
```
DecisionTreeClassifier(max_depth=5, min_samples_split=10)
```

In [95]:
```
ypred_dt = dt.predict(x_test)
print(ypred_dt)
```

```
[0 0 1 ... 1 0 0]
```

In [96]:
```
eval_model(y_test, ypred_dt)
```

```
Accuracy Score 0.9087378640776699

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [96], in <cell line: 1>()
----> 1 eval_model(y_test, ypred_dt)

Input In [89], in eval_model(y_test, y_pred)
      3 print('Accuracy Score', acc)
      4 cm = confusion_matrix(y_test,y_pred)
----> 5 print('Confusion Matrix\n', cn)
      6 print('Classification Report\n', classification_report(y_test,y_pred))

NameError: name 'cn' is not defined
```

In [97]:
```
from sklearn import tree, plot_tree
```

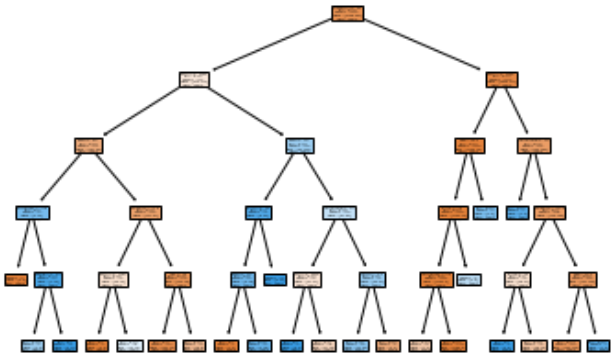
In [98]:
```
cn =[ 'no', 'yes']
fn = x_train.columns
```

In [97]:
```
from sklearn.tree import plot_tree
```

In [98]:
```
cn =[ 'no', 'yes']
fn = x_train.columns
print(fn)
print(cn)
```

```
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'],
      dtype='object')
['no', 'yes']
```

In [100]:
```
feature_names = df.columns.tolist()
class_names = ["class_0", "class_1"]
plot_tree(dt, feature_names=feature_names, class_names=class_names, filled=True)
plt.show()
```



In [102]:
```
dt1 = DecisionTreeClassifier(criterion='entropy', max_depth=4,min_samples_split=1
dt1.fit(x_train,y_train)
```

Out[102]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=4, min_samples_split=15)
```

In [103]:
```
mscore(dt1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [103], in <cell line: 1>()
----> 1 mscore(dt1)

NameError: name 'mscore' is not defined
```

In [104]:
```
ypred_dt1 = dt1.predict(x_test)
```

In [105]:
```
eval_model(y_test,ypred_dt1)
```

```
Accuracy Score 0.9106796116504854
Confusion Matrix
 ['no', 'yes']
```

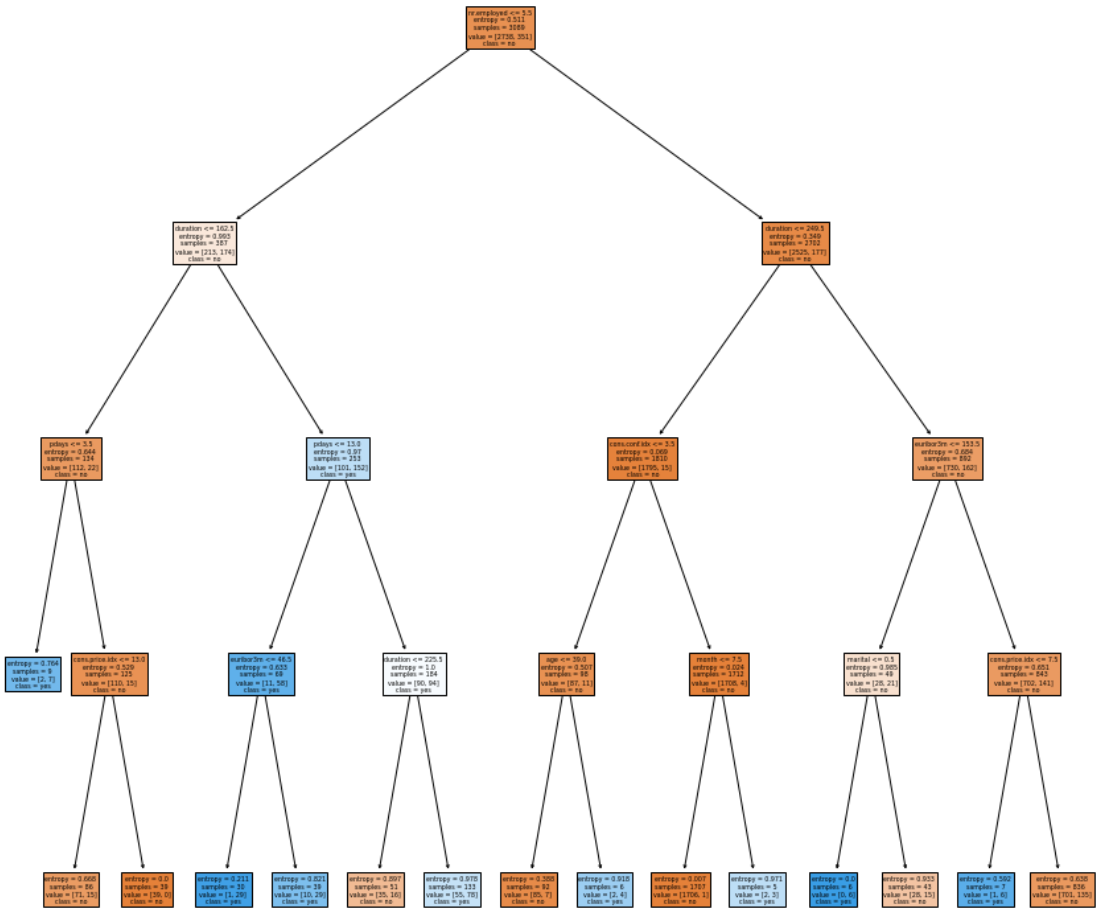
In [105]:
```
eval_model(y_test,ypred_dt1)
```

```
Accuracy Score 0.9106796116504854
Confusion Matrix
 ['no', 'yes']
Classification Report
              precision    recall  f1-score   support

           0       0.94      0.96      0.95       930
           1       0.55      0.42      0.48       100

    accuracy                           0.91      1030
   macro avg       0.75      0.69      0.71      1030
weighted avg       0.90      0.91      0.91      1030
```

In [106]:
```
plt.figure(figsize=(15, 15))
plot_tree(dt1, feature_names=fn.tolist(), class_names=cn, filled=True)
plt.show()
```



In [ ]:
```

```

In [ ]: