

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

```
In [2]: bank=pd.read_excel("C:\\Users\\Pranav\\Desktop\\DATA SCIENCE DATA\\Excel file\\bank-full.xlsx")
bank.head()
```

Out[2]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
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

```
In [3]: # shape of dataset
bank.shape
```

Out[3]: (45211, 17)

```
In [4]: # information of dataset
bank.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   age         45211 non-null  int64
 1   job         45211 non-null  object
 2   marital     45211 non-null  object
 3   education   45211 non-null  object
 4   default     45211 non-null  object
 5   balance     45211 non-null  int64
 6   housing     45211 non-null  object
 7   loan        45211 non-null  object
 8   contact     45211 non-null  object
 9   day         45211 non-null  int64
10  month       45211 non-null  object
11  duration    45211 non-null  int64
12  campaign    45211 non-null  int64
13  pdays       45211 non-null  int64
14  previous    45211 non-null  int64
15  poutcome    45211 non-null  object
16  y           45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

```
In [5]: # describe the matamatical statagic of data
bank.describe()
```

Out[5]:

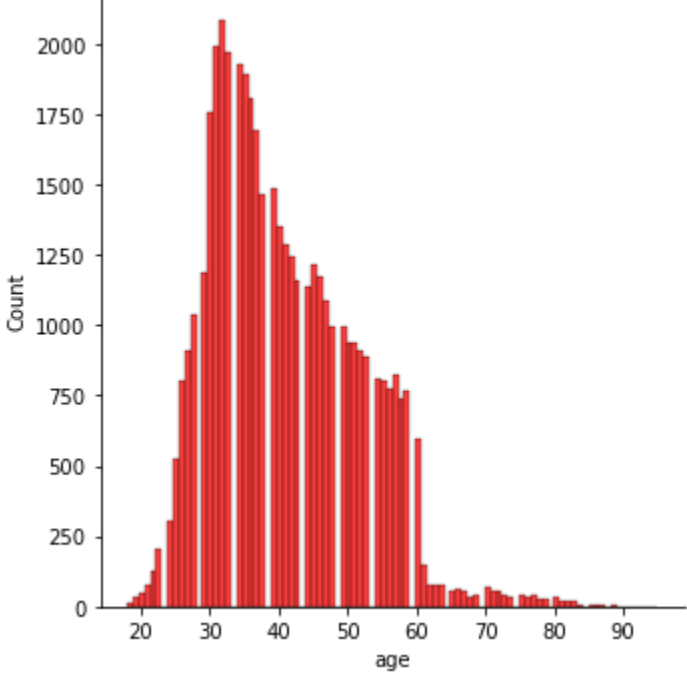
	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

```
In [6]: # Finding the missing value in dataset
bank.isnull().sum()
```

Out[6]:

```
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
y            0
dtype: int64
```

```
In [7]: # plotting the distplot
sns.distplot(bank['age'],color='red')
plt.show()
```



```
In [8]: # measuring the value in job columns
bank['job'].value_counts()
```

Out[8]:

```
blue-collar    9732
management    9458
technician     7597
admin.         5171
services       4154
retired        2264
self-employed  1579
entrepreneur   1487
unemployed     1303
housemaid      1240
student        938
unknown        288
Name: job, dtype: int64
```

```
In [9]: # measuring the value in marital columns
bank['marital'].value_counts()
```

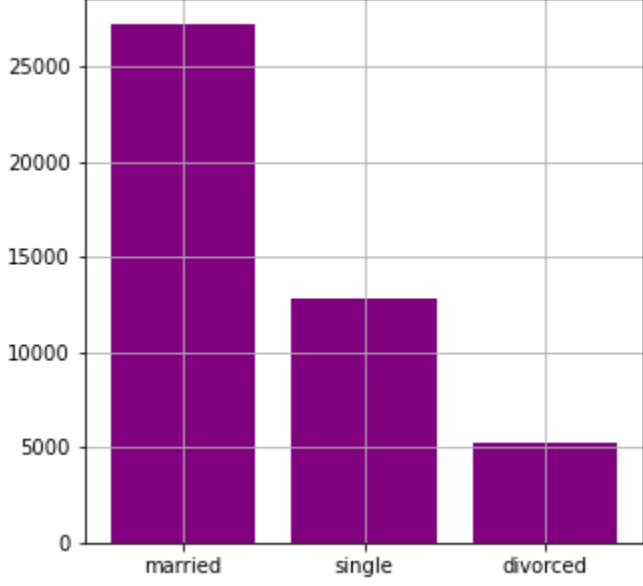
Out[9]:

```
married    27214
single     12790
divorced   5207
Name: marital, dtype: int64
```

```
In [10]: # BarPlot
plt.figure(figsize = (16, 5));
plt.bar(list(bank['job'].value_counts().keys()),list(bank['job'].value_counts()),color='pink')
plt.grid(True)
plt.show()
```



```
In [11]: plt.figure(figsize = (5, 5));
plt.bar(list(bank['marital'].value_counts().keys()),list(bank['marital'].value_counts()),color='purple')
plt.grid(True)
plt.show()
```

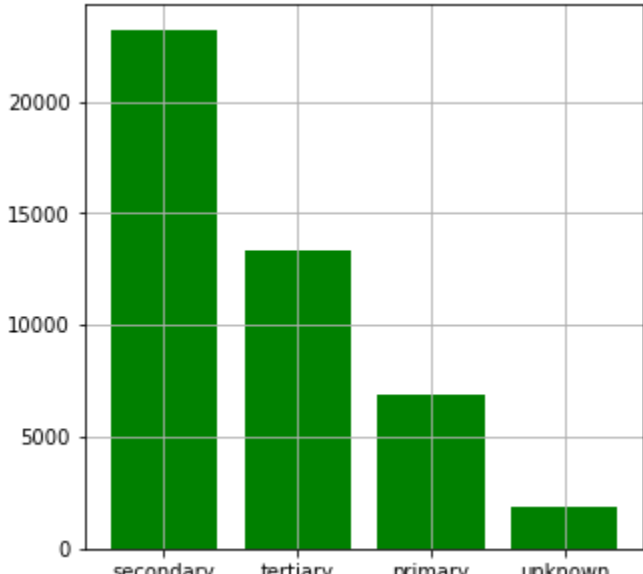


```
In [12]: # measuring value in education
bank['education'].value_counts()
```

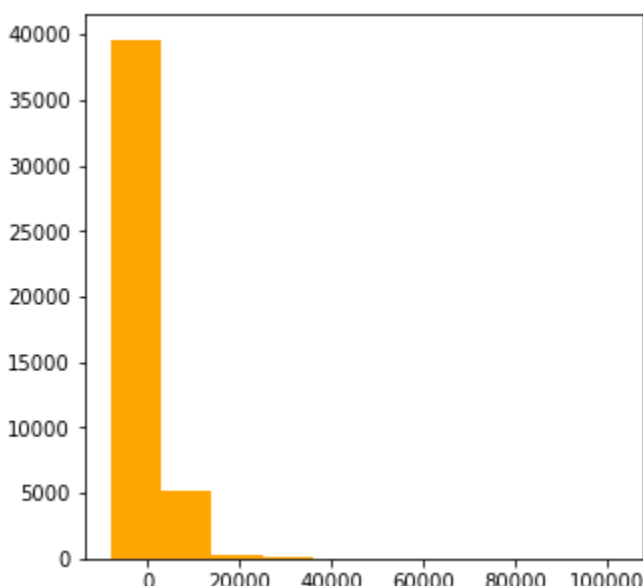
Out[12]:

```
secondary    23202
tertiary     13301
primary       6851
unknown       1857
Name: education, dtype: int64
```

```
In [13]: plt.figure(figsize = (5, 5));
plt.bar(list(bank['education'].value_counts().keys()),list(bank['education'].value_counts()),color='green')
plt.grid(True)
plt.show()
```



```
In [14]: plt.figure(figsize = (5, 5));
plt.hist(bank['balance'],color='orange')
plt.show()
```

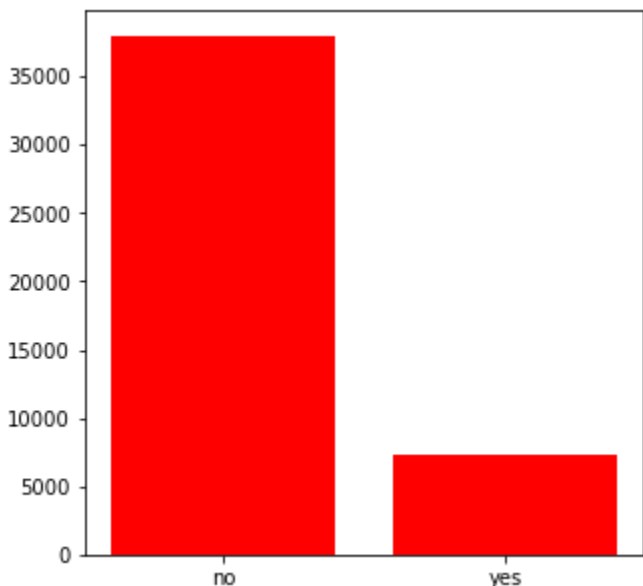


```
In [15]: #measuring the value in loan
bank['loan'].value_counts()
```

Out[15]:

```
no    37967
yes    7244
Name: loan, dtype: int64
```

```
In [16]: plt.figure(figsize = (5, 5));
plt.bar(list(bank['loan'].value_counts().keys()),list(bank['loan'].value_counts()),color='red')
plt.show()
```



```
In [22]: x=bank['age'].values.reshape(-1, 1)
y=bank['balance'].values.reshape(-1, 1)
```

```
In [23]: x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.3)
print("shape of x_train= ",x_train.shape)
print("shape of x_test= ",x_test.shape)
print("shape of y_train= ",y_train.shape)
print("shape of y_test= ",y_test.shape)
```

```
shape of x_train= (31647, 1)
shape of x_test= (13564, 1)
shape of y_train= (31647, 1)
shape of y_test= (13564, 1)
```

```
In [24]: rfg=RandomForestClassifier()
```

```
In [25]: rfg.fit(x_train,y_train)

C:\Users\Pranav\AppData\Local\Temp\ipykernel_24500\2097697493.py:1: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,) for example using.ravel().
  rfg.fit(x_train,y_train)
```

Out[25]:

```
▼ RandomForestClassifier
RandomForestClassifier()
```

```
In [28]: y_pred=rfg.predict(x_test)
```

```
In [34]: y_pred[0:5],y_test[0:5]
```

Out[34]:

```
(array([0, 0, 0, 0, 0], dtype=int64),
 array([[ 781],
        [  0],
        [ 134],
        [ 877]], dtype=int64))
```

```
In [35]: from sklearn.metrics import mean_squared_error
```

```
In [36]: mean_squared_error(y_test,y_pred)
```

Out[36]: 10880016.04924801

```
In [ ]:
```