```
# IMPORTANT: SOME KAGGLE DATA SOURCES ARE PRIVATE
# RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES.
import kagglehub
kagglehub.login()
```

```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES,
# TO ATTACH A DATA SOURCE TO THIS NOTEBOOK
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.

kagglehub_bank_dir_path = kagglehub.dataset_download('kagglehub/bank-full-versions')
kagglehub_bank_full_versions_path = kagglehub.dataset_download('kagglehub/bank-full-versions')

print('Data source import complete.')
```

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/bank-45k/bank.csv
/kaggle/input/bank-full-versions/bank-full.csv
```

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

Problem Description:

ABC Bank wants to sell it's term deposit product to customers and before launching the product they want to develop a model which help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

).

Business Understanding:

Bank wants to use ML model to shortlist customer whose chances of buying the product is more so that their marketing channel (tele marketing, SMS/email marketing etc) can focus only to those customers whose chances of buying the product is more.This will save resource and their time (which is directly involved in the cost (resource billing)).Develop model with Duration and without duration feature and report the performance of the model.Duration feature is not recommended as this will be difficult to explain the result to business and also it will be difficult for business to campaign based on duration.

Data Set Information :

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (yes') or not ('no') subscribed.

The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

Attribute Information:

input variables:

bank client data:

1 - age (numeric)

2 - job :type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')

3 - marital: marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed

4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')

5 - default: has credit in default? (categorical: 'no','yes','unknown')

6 - housing: has housing loan? (categorical: 'no','yes','unknown'

7 - loan: has personal loan? (categorical: 'no','yes','unknown')

related with the last contact of the current campaign:

8 - contact: contact communication type (categorical: 'cellular','telephone'

9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec'

10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri'

11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

other attributes:

12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted

14 - previous: number of contacts performed before this campaign and for this client (numeric)

15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

social and economic context attribute

16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)

17 - cons.price.idx: consumer price index - monthly indicator (numeric)

18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)

19 - euribor3m: euribor 3 month rate - daily indicator (numeric

20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):

21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

e) 1e0

```
Bank_data = pd.read_csv("/kaggle/input/bank-full-versions/bank-full.csv")
```

```
import pandas as pd

# Load your CSV file
Bank_data = pd.read_csv('/kaggle/input/bank-full-versions/bank-full.csv')

# Display column names
print(Bank_data.columns)
```

```
Index(['sl. no', 'age', 'job', 'marital', 'education', 'default', 'balance',
       'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign',
       'pdays', 'previous', 'poutcome', 'y'],
      dtype='object')
```

```
Bank_data.head()
```

	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 × 18 columns

```
Bank_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 18 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   sl. no     45211 non-null  int64
 1   age        45211 non-null  int64
 2   job        45211 non-null  object
 3   marital    45211 non-null  object
 4   education  45211 non-null  object
 5   default    45211 non-null  object
 6   balance    45211 non-null  int64
 7   housing    45211 non-null  object
 8   loan       45211 non-null  object
 9   contact    45211 non-null  object
 10  day        45211 non-null  int64
 11  month      45211 non-null  object
 12  duration   45211 non-null  int64
 13  campaign   45211 non-null  int64
 14  pdays      45211 non-null  int64
 15  previous   45211 non-null  int64
 16  poutcome   45211 non-null  object
 17  y          45211 non-null  object
dtypes: int64(7), object(11)
memory usage: 6.2+ MB
```

Check rating in descending with AI

Double-click (or enter) to edit

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 × 18 columns

Handle missing values

```
# Check for missing values in all datasets
print(Bank_data.isnull().sum())
```

```
sl. no       0
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
```

Exploratory Data Analysis (EDA)

```
# Importing the external libraries
import pandas as pd

# Importing the data
Bank_dataspd.read_csv("/kaggle/input/bank-full-versions/bank-full.csv")
```

```
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 × 18 columns

```
new_bank_data = Bank_data.sort_values(by='balance', ascending=True)
```

```
#Outlier Detection and Handling
#Identify outliers in numerical columns, e.g. the 'balance' column:
Q1 = new_bank_data['balance'].quantile(0.25)
Q3 = new_bank_data['balance'].quantile(0.75)
IQR= Q3-Q1

print(Q1)
print(Q3)
print(IQR)
```

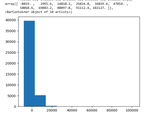
```
72.0
1428.0
1356.0
```

```
Bank_data=Bank_data[~((Bank_data['balance']<(Q1-1.5*IQR))|(Bank_data['balance']>(Q3+1.5*IQR)))]
print(Bank_data)
```

```
       sl. no  age           job  marital  education default  balance  \
0           1   58    management  married   tertiary      no     2143
5           6   35    management   single   tertiary      no      231
7           8   23       student   single    unknown      no      113
8           9   28    management   single   tertiary      no      328
...       ...  ...           ...      ...        ...     ...      ...
45186   45186   25    technician   single  secondary      no      305
45187   45187   51    technician  married  secondary      no      825
45207   45208   71       retired  divorced    primary      no     1729
45208   45209   72       retired  married  secondary      no      668
45210   45211   37  entrepreneur  married  secondary      no     2971

       housing loan   contact  day  month  duration  campaign  pdays  previous  \
0          yes   no   unknown    5    may       261         1     -1         0
5          yes   no   unknown    5    may       139         1     -1         0
7           no   no   unknown    5    may       274         1     -1         0
8          yes   no   unknown    5    may        92         1     -1         0
...        ...  ...       ...  ...    ...       ...       ...    ...       ...
45186       no  yes  cellular   17    nov       386         2     -1         0
45187       no   no  cellular   17    nov       977         3     -1         0
45207       no   no  cellular   17    nov       456         2     -1         0
45208       no   no  telephone  17    nov       508         4     -1         0
45210       no   no  cellular   17    nov       361         2    188        11

       poutcome   y
0       unknown  no
5       unknown  no
7       unknown  no
8       unknown  no
...         ...  ..
45186   unknown  no
45187   unknown  no
45207   unknown  no
45208   unknown  no
45210     other  no

[29468 rows x 18 columns]
```

```
# Histogram
import matplotlib.pyplot as plt
plt.hist(Bank_data['balance'])
```

```
(array([2.9468e+04, 1.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 1.0000e+00]),
 array([   -8019. ,  -794.1,  6430.8, 13655.7, 20880.6, 28105.5, 35330.4,
        42555.3, 49780.2, 57005.1, 64230. ]),
 <BarContainer object of 10 artists>)
```



```
import pandas as pd

Bank_dataspd.read_csv("/kaggle/input/bank-full-versions/bank-full.csv")
```

```
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
5	6	35	management	single	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no
7	8	23	student	single	unknown	no	113	no	no	unknown	5	may	274	1	-1	0	unknown	no
8	9	28	management	single	tertiary	no	328	yes	no	unknown	5	may	92	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45186	45186	25	technician	single	secondary	no	305	yes	no	cellular	17	nov	386	2	-1	0	unknown	yes
45187	45187	51	technician	married	secondary	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	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

29468 rows × 18 columns

```
# Importing the external libraries
import pandas as pd

# Importing the data
Bank_dataspd.read_csv("/kaggle/input/bank-full-versions/bank-full.csv")
```

```
Bank_data.isnull().sum()
```

```
sl. no       0
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
```

```
Bank_data.columns
```

```
Index(['sl. no', 'age', 'job', 'marital', 'education', 'default', 'balance',
       'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign',
       'pdays', 'previous', 'poutcome', 'y'],
      dtype='object')
```

```
Bank_data.values
```

```
array([[1, 58, 'management', ..., 0, 'unknown', 'no'],
       [6, 35, 'management', ..., 0, 'unknown', 'no'],
       [8, 23, 'entrepreneur', ..., 0, 'unknown', 'no'],
       ...,
       [45208, 72, 'retired', ..., 3, 'success', 'yes'],
       [45209, 57, 'blue-collar', ..., 0, 'unknown', 'no'],
       [45210, 37, 'entrepreneur', ..., 11, 'other', 'no']], dtype=object)
```

```
Bank_data.dtypes
```

```
sl. no        int64
age           int64
job          object
marital      object
education    object
default      object
balance       int64
housing      object
loan         object
contact      object
day           int64
month        object
duration      int64
campaign      int64
pdays         int64
previous      int64
poutcome     object
y            object
dtype: object
```

```
Bank_data.describe()
```

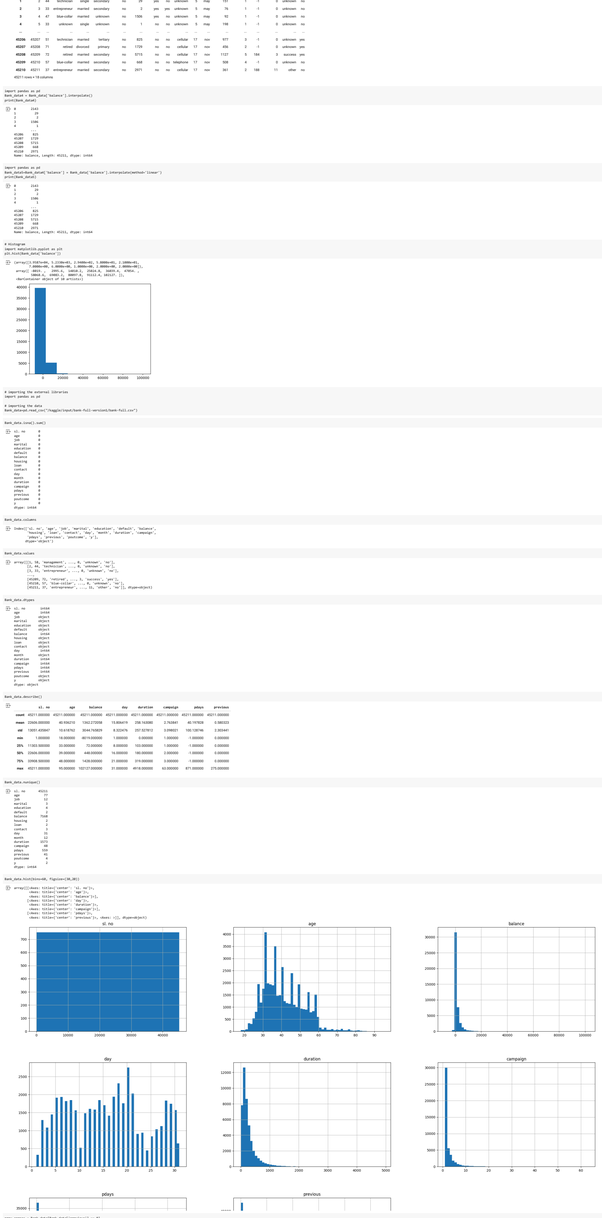
	sl. no	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	22606.000000	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	13057.420941	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	1.000000	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	11303.500000	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	22606.000000	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	33908.500000	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	45211.000000	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

```
Bank_data.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
dtype: int64
```

```
Bank_data.hist(bins=50, figsize=(30,30))
```

```
array([[<Axes: title={'center': 'sl. no'}>,
        <Axes: title={'center': 'age'}>,
        <Axes: title={'center': 'balance'}>],
       [<Axes: title={'center': 'day'}>,
        <Axes: title={'center': 'duration'}>,
        <Axes: title={'center': 'campaign'}>],
       [<Axes: title={'center': 'pdays'}>,
        <Axes: title={'center': 'previous'}>, <Axes: >]], dtype=object)
```




```
prev_nonzero = Bank_data[Bank_data['previous'] != 0]
```

```
prev_nonzero['poutcome'].unique()
```

```
array(['unknown', 'other', 'success', 'nonexistent', 'failure'],
      dtype=object)
```

```
prev_zeros = Bank_data[Bank_data['previous'] = 0]
```

```
prev_zeros['poutcome'].unique()
```

```
array(['failure', 'other', 'success', 'unknown'], dtype=object)
```

```
Bank_DataFr = Bank_data.loc[(Bank_data['housing'] == 'no') & (Bank_data['loan'] == 'no') & (Bank_data['contact'] == 'telephone')]
```

```
Bank_DataFr
```

	sl. no	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
12660	12661	31	technician	single	secondary	no	19	no	no	telephone	4	jul	65	2	-1	0	unknown	no
12665	12666	25	blue-collar	single	secondary	no	192	no	no	telephone	4	jul	112	2	-1	0	unknown	no
12674	12675	36	unemployed	divorced	primary	no	0	no	no	telephone	4	jul	304	3	-1	0	unknown	no
12716	12717	49	technician	married	primary	no	486	no	no	telephone	7	jul	229	5	-1	0	unknown	no
12750	12751	46	unknown	married	unknown	no	2911	no	no	telephone	7	jul	41	9	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45141	45142	77	unknown	married	unknown	no	397	no	no	telephone	8	nov	207	1	185	3	success	no
45154	45155	63	retired	married	primary	no	3738	no	no	telephone	9	nov	301	4	656	4	failure	no
45166	45167	41	unemployed	single	tertiary	no	79	no	no	telephone	9	nov	394	1	390	2	success	yes
45170	45171	19	student	single	primary	no	245	no	no	telephone	10	nov	98	2	110	2	other	no
45209	45210	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no

1509 rows × 18 columns

```
Bank_DataFr['default'].unique()
```

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

```
# importing the external libraries
import pandas as pd

# importing the data
Bank_data=pd.read_csv("/kaggle/input/bank-full-version1/bank-full.csv")
```

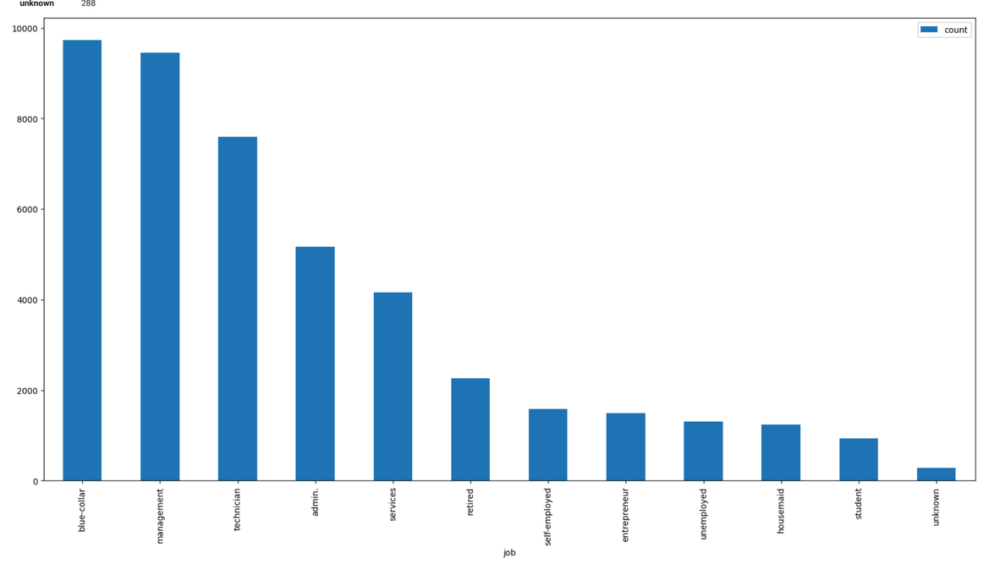
```
Bank_data
```

	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 × 18 columns

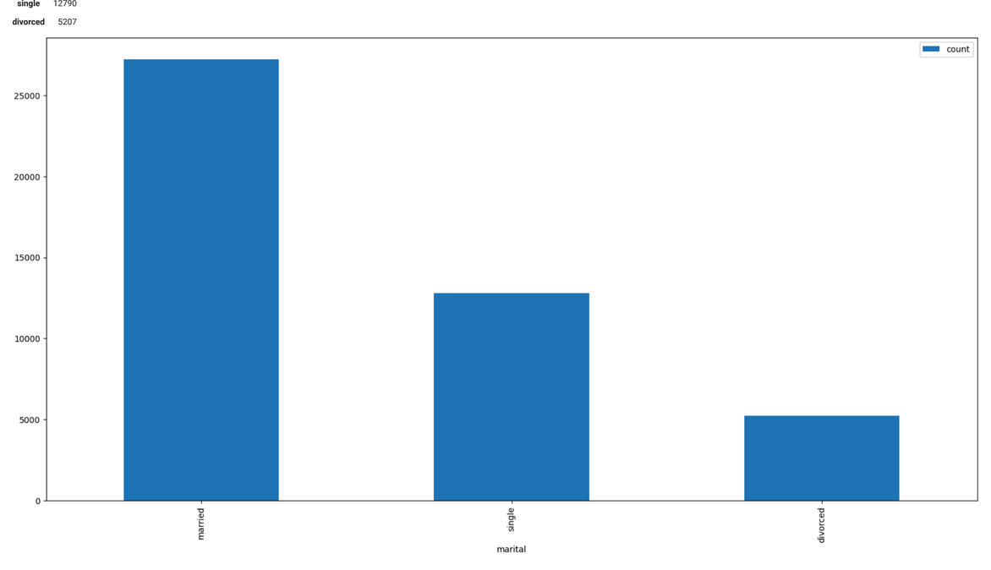
```
pd.DataFrame(Bank_data['job'].value_counts()).plot(kind='bar', figsize=(20,10))
pd.DataFrame(Bank_data['job'].value_counts())
```

job	count
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




```
pd.DataFrame(Bank_data['marital'].value_counts()).plot(kind='bar', figsize=(20,10))
pd.DataFrame(Bank_data['marital'].value_counts())
```

marital	count
married	27214
single	12790
divorced	5207




```
newDataFrame = Bank_data
```

```
newDataFrame['default'] = newDataFrame['default'].replace('unknown', np.nan)
```

```
newDataFrame['loan'] = newDataFrame['loan'].replace('unknown', np.nan)
```

```
newDataFrame['housing'] = newDataFrame['housing'].replace('unknown', np.nan)
```

```
newDataFrame['education'] = newDataFrame['education'].replace('unknown', np.nan)
```

```
newDataFrame['job'] = newDataFrame['job'].replace('unknown', np.nan)
```

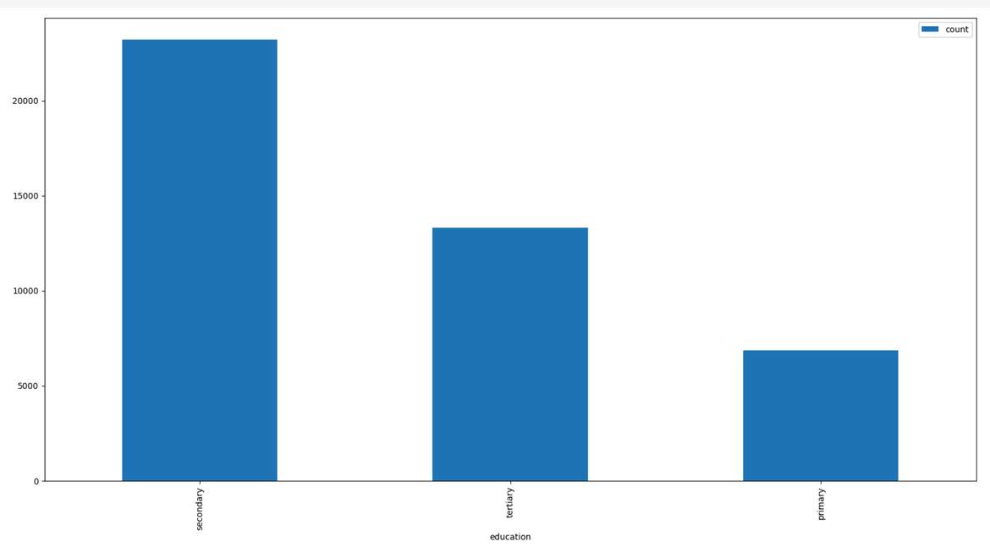
```
newDataFrame['marital'] = newDataFrame['marital'].replace('unknown', np.nan)
```

```
newDataFrame
```

	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	NaN	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	5	33	NaN	single	NaN	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 × 18 columns

```
import pandas as pd
import matplotlib.pyplot as plt

# Assuming newDataFrame is already defined
pd.DataFrame(newDataFrame['education'].value_counts()).plot(kind='bar', figsize=(20, 10))

# Display the plot
plt.show()
```




```
newDataFrame.isnull().sum()
```

```
sl. no          0
age             0
job           288
marital         0
education    1857
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
```

```
newDataFrame1 = newDataFrame.sort_values(by='age', ascending=True)
```

```
newDataFrame1
```

	sl. no	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
42954	42955	18	student	single	NaN	no	108	no	no	cellular	9	feb	92	1	183	1	success	yes
41273	41274	18	student	single	NaN	no	3	no	no	cellular	25	aug	130	2	-1	0	unknown	yes
42146	42147	18	student	single	secondary	no	156	no	no	cellular	4	nov	298	2	82	4	other	no
43637	43638	18	student	single	NaN	no	348	no	no	cellular	5	may	443	4	-1	0	unknown	yes
40736	40737	18	student	single	primary	no	1944	no	no	telephone	10	aug	122	3	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
44420	44421	93	retired	married	NaN	no	775	no	no	cellular	4	aug	476	2	13	9	success	yes
44262	44263	93	retired	married	NaN	no	775	no	no	cellular	22	jul	860	3	177	7	success	yes
31233	31234	94	retired	divorced	secondary	no	1234	no	no	cellular	3	mar	212	1	-1	0	unknown	no
41663	41664	95	retired	married	secondary	no	0	no	no	telephone	1	oct	215	1	-1	0	unknown	no
33699	33700	95	retired	divorced	primary	no	2282	no	no	telephone	21	apr	207	17	-1	0	unknown	yes

45211 rows × 18 columns

```
newDataFrame['job'].unique()
```

```
array(['management', 'technician', 'entrepreneur', 'blue-collar', nan,
       'retired', 'admin.', 'services', 'self-employed', 'unemployed',
       'housemaid', 'student'], dtype=object)
```

```
data_marital = pd.DataFrame()

data_marital['student'] = newDataFrame[newDataFrame['job'] == 'student']['marital'].value_counts()
data_marital['blue-collar'] = newDataFrame[newDataFrame['job'] == 'blue-collar']['marital'].value_counts()
data_marital['admin.'] = newDataFrame[newDataFrame['job'] == 'admin.']['marital'].value_counts()
data_marital['services'] = newDataFrame[newDataFrame['job'] == 'services']['marital'].value_counts()
data_marital['unemployed'] = newDataFrame[newDataFrame['job'] == 'unemployed']['marital'].value_counts()
data_marital['technician'] = newDataFrame[newDataFrame['job'] == 'technician']['marital'].value_counts()
data_marital['retired'] = newDataFrame[newDataFrame['job'] == 'retired']['marital'].value_counts()
data_marital['management'] = newDataFrame[newDataFrame['job'] == 'management']['marital'].value_counts()
data_marital['entrepreneur'] = newDataFrame[newDataFrame['job'] == 'entrepreneur']['marital'].value_counts()
data_marital['self-employed'] = newDataFrame[newDataFrame['job'] == 'self-employed']['marital'].value_counts()
data_marital['housemaid'] = newDataFrame[newDataFrame['job'] == 'housemaid']['marital'].value_counts()
data_marital['retired'] = newDataFrame[newDataFrame['job'] == 'retired']['marital'].value_counts()
data_marital['nan'] = newDataFrame[newDataFrame['job'] == 'nan']['marital'].value_counts()
```

```
data_marital
```

marital	student	blue-collar	admin.	services	unemployed	technician	retired	management	entrepreneur	self-employed	housemaid	nan
single	878	2014	1728	1198	401	2620	108	2947	238	446	144	NaN
married	54	6968	2693	2407	731	4052	1731	5400	1070	993	912	NaN
divorced	6	750	750	549	171	925	425	1111	179	140	184	NaN

```
newDataFrame['marital'].unique()
```

```
array(['married', 'single', 'divorced'], dtype=object)
```

```
import pandas as pd


agetomarital = pd.DataFrame()
agetomarital['married'] = newDataFrame[newDataFrame['marital'] == 'married']['age'].value_counts()
agetomarital['single'] = newDataFrame[newDataFrame['marital'] == 'single']['age'].value_counts()
agetomarital['divorced'] = newDataFrame[newDataFrame['marital'] == 'divorced']['age'].value_counts()

print(agetomarital)
```

```
     married  single  divorced
age
34      1131   650.0     149.0
35      1077   631.0     186.0
36      1076   538.0     192.0
33      1075   746.0     151.0
37      1073   453.0     170.0
..       ...     ...       ...
89         2     NaN       1.0
92         2     NaN       NaN
93         2     NaN       NaN
88         2     NaN       NaN
95         1     NaN       1.0

[73 rows x 3 columns]
```

```
agetomarital.sort_index()
```

age	married	single	divorced
20	3	47.0	NaN
21	5	74.0	NaN
22	9	120.0	NaN
23	27	175.0	NaN
24	53	248.0	1.0
...	...	...	...
88	2	NaN	NaN
89	2	NaN	1.0
92	2	NaN	NaN
93	2	NaN	NaN
95	1	NaN	1.0

73 rows × 3 columns

```
newDataFrame['education'].unique()
```

```
array(['tertiary', 'secondary', nan, 'primary'], dtype=object)
```

```
import pandas as pd


agetoeducation = pd.DataFrame()
agetoeducation['tertiary'] = newDataFrame[newDataFrame['education'] == 'tertiary']['age'].value_counts()
agetoeducation['secondary'] = newDataFrame[newDataFrame['education'] == 'secondary']['age'].value_counts()
agetoeducation['unknown'] = newDataFrame[newDataFrame['education'] == 'unknown']['age'].value_counts()
agetoeducation['primary'] = newDataFrame[newDataFrame['education'] == 'primary']['age'].value_counts()

print(agetoeducation)
```

```
     tertiary  secondary  unknown  primary
age
32        776     1088.0      NaN      179
33        760     1011.0      NaN      167
31        749     1065.0      NaN      139
34        718     1000.0      NaN      166
30        686      913.0      NaN      124
..        ...        ...      ...      ...
82          2        3.0      NaN       12
81          2        3.0      NaN       10
80          1       12.0      NaN       17
89          1        NaN      NaN        2
84          1        3.0      NaN        5

[65 rows x 4 columns]
```

```
agetoeducation.sort_index()
```

age	tertiary	secondary	unknown	primary
20	3	26.0	NaN	6
21	6	54.0	NaN	10
22	14	96.0	NaN	8
23	33	143.0	NaN	17
24	44	219.0	NaN	19
...	...	...	...	...
81	2	3.0	NaN	10
82	2	3.0	NaN	12
83	5	2.0	NaN	14
84	1	3.0	NaN	5
89	1	NaN	NaN	2

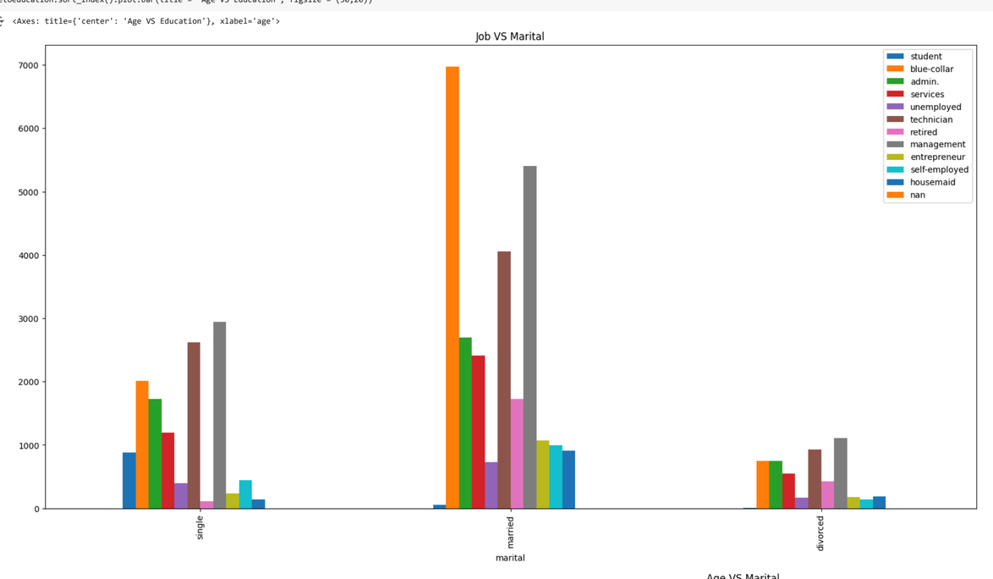
65 rows × 4 columns

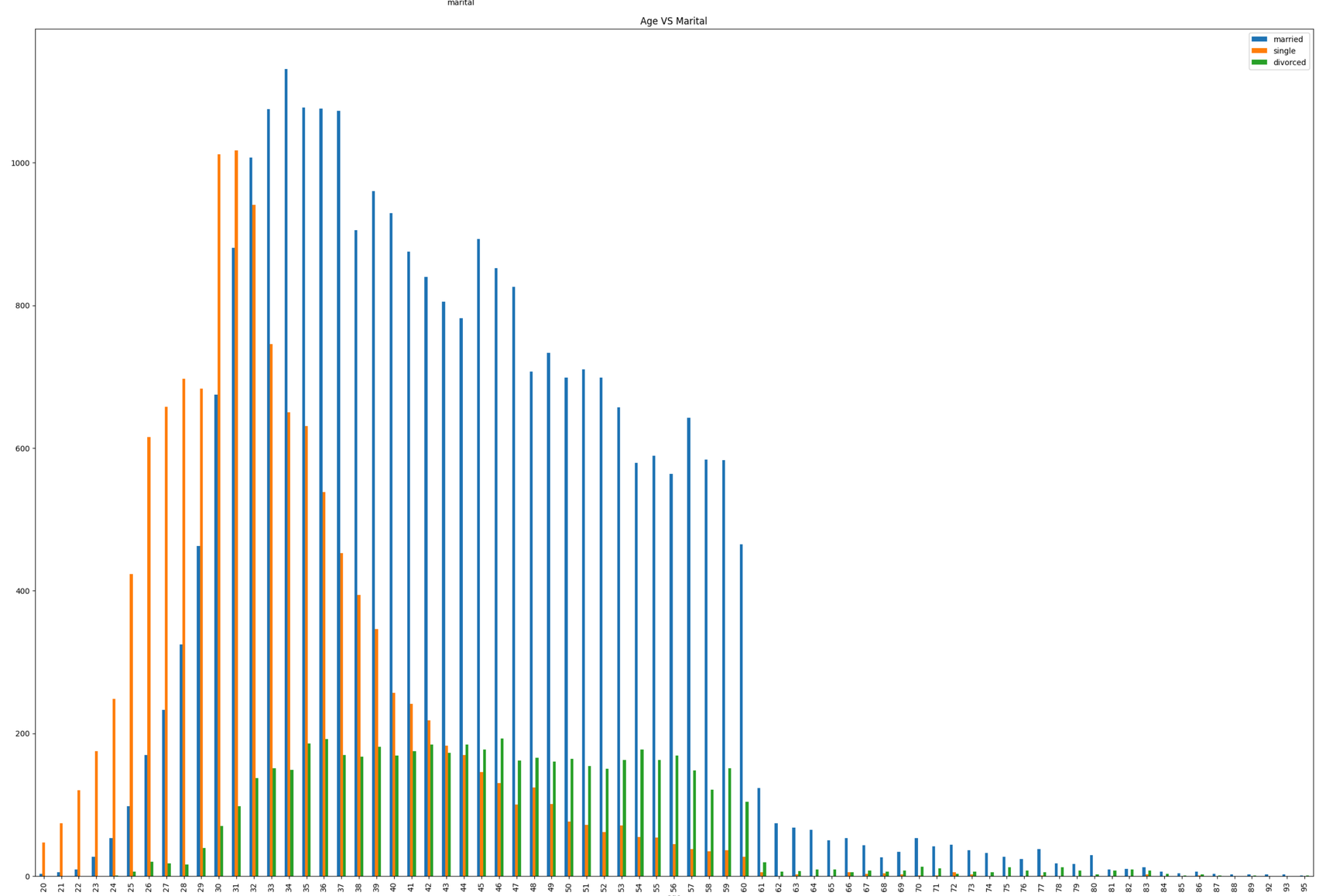
```
print(data_marital.columns)
```

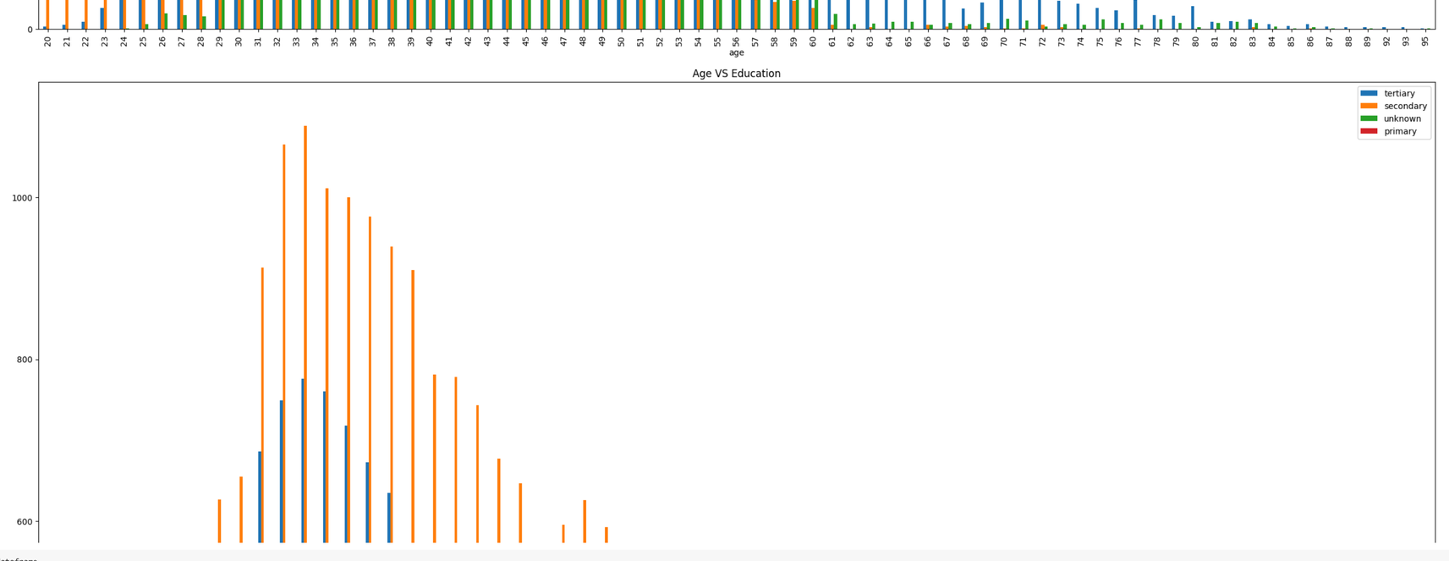
```
Index(['student', 'blue-collar', 'admin.', 'services', 'unemployed',
       'technician', 'retired', 'management', 'entrepreneur', 'self-employed',
       'housemaid', 'nan'],
      dtype='object')
```

```
data_marital.plot.bar(title = "Job VS Marital", figsize=(20,10))
agetomarital.sort_index().plot.bar(title = 'Age VS Marital', figsize = (30,20))
agetoeducation.sort_index().plot.bar(title = 'Age VS Education', figsize = (30,20))
```

```
<Axes: title={'center': 'Age VS Education'}, xlabel='age'>
```








```
newDataFrame
```

	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	NaN	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	5	33	NaN	single	NaN	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 × 18 columns

```
import pandas as pd


agetodeposit = pd.DataFrame()
agetodeposit['yes'] = newDataFrame[newDataFrame['y'] == 'yes']['age'].value_counts()
agetodeposit['no'] = newDataFrame[newDataFrame['y'] == 'no']['age'].value_counts()

print(agetodeposit)
```

```
     yes      no
age
32   221  1864.0
30   217  1540.0
33   210  1762.0
35   209  1685.0
31   206  1790.0
..   ...     ...
87     3     1.0
90     2     NaN
92     2     NaN
93     2     NaN
95     1     1.0

[74 rows x 2 columns]
```

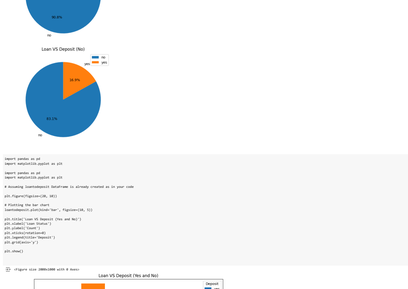
```
import pandas as pd
import matplotlib.pyplot as plt


agetodeposit = pd.DataFrame()
agetodeposit['yes'] = newDataFrame[newDataFrame['y'] == 'yes']['age'].value_counts()
agetodeposit['no'] = newDataFrame[newDataFrame['y'] == 'no']['age'].value_counts()

# Plotting the line plot
agetodeposit.sort_index().plot(kind='line', title="Age VS Deposit", figsize=(20, 10))
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```

```
agetodeposit = pd.DataFrame()

agetodeposit['yes'] = newdataframe[newdataframe['y'] == 'yes']['age'].value_counts()
agetodeposit['no'] = newdataframe[newdataframe['y'] == 'no']['age'].value_counts()

agetodeposit.sort_index().plot.bar(title = "Age VS Deposit", figsize=(30,10))
```

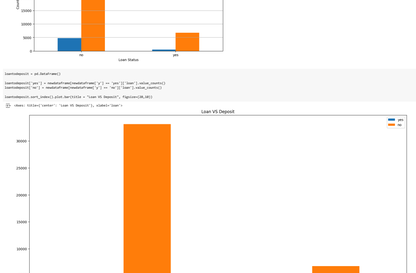
```
<Axes: title={'center': 'Age VS Deposit'}, xlabel='Age'>
```


```
jobtodeposit = pd.DataFrame()

jobtodeposit['yes'] = newdataframe[newdataframe['y'] == 'yes']['job'].value_counts()
jobtodeposit['no'] = newdataframe[newdataframe['y'] == 'no']['job'].value_counts()

jobtodeposit.sort_index().plot.bar(title = "Job VS Deposit", figsize=(30,10))
```

```
<Axes: title={'center': 'Job VS Deposit'}, xlabel='Job'>
```

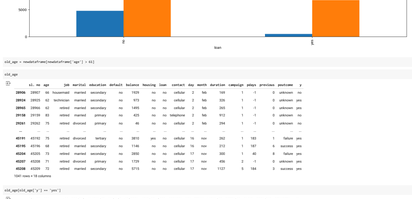



```
import pandas as pd
import matplotlib.pyplot as plt


housingtodeposit = pd.DataFrame()
housingtodeposit['yes'] = newdataframe[newdataframe['y'] == 'yes']['housing'].value_counts()
housingtodeposit['no'] = newdataframe[newdataframe['y'] == 'no']['housing'].value_counts()

# Plotting the stacked bar chart
housingtodeposit.plot(kind='bar', stacked=True, title='Housing VS Deposit', figsize=(20, 10))
plt.xlabel('Housing')
plt.ylabel('Count')
plt.show()
```

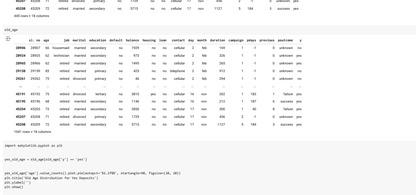



```
housingtodeposit = pd.DataFrame()

housingtodeposit['yes'] = newdataframe[newdataframe['y'] == 'yes']['housing'].value_counts()
housingtodeposit['no'] = newdataframe[newdataframe['y'] == 'no']['housing'].value_counts()

housingtodeposit.sort_index().plot.bar(title = "Housing VS Deposit", figsize=(30,10))
```

```
<Axes: title={'center': 'Housing VS Deposit'}, xlabel='housing'>
```

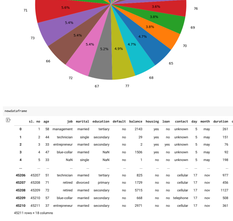



```
import pandas as pd
import matplotlib.pyplot as plt


loantodeposit = pd.DataFrame()
loantodeposit['yes'] = newdataframe[newdataframe['y'] == 'yes']['loan'].value_counts()
loantodeposit['no'] = newdataframe[newdataframe['y'] == 'no']['loan'].value_counts()

plt.figure(figsize=(20, 10))

# Plotting the pie chart
loantodeposit['yes'].plot.pie(autopct='%1.1f%%', startangle=90, title='Loan VS Deposit (Yes)', figsize=(10, 5), legend=True)
plt.ylabel('') # Hide y-label
plt.show()

plt.figure(figsize=(20, 10))
loantodeposit['no'].plot.pie(autopct='%1.1f%%', startangle=90, title='Loan VS Deposit (No)', figsize=(10, 5), legend=True)
plt.ylabel('') # Hide y-label
plt.show()
```



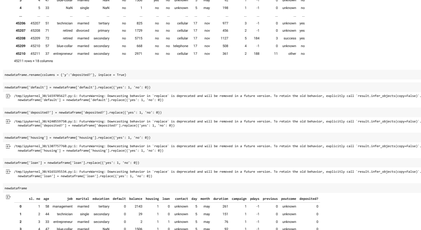

```
import pandas as pd
import matplotlib.pyplot as plt

import pandas as pd
import matplotlib.pyplot as plt

# Assuming loantodeposit DataFrame is already created as in your code

plt.figure(figsize=(20, 10))

# Plotting the bar chart
loantodeposit.plot(kind='bar', figsize=(10, 5))

plt.title('Loan VS Deposit (Yes and No)')
plt.xlabel('Loan Status')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Deposit')
plt.grid(axis='y')

plt.show()
```

```
<Figure size 2000x1000 with 0 Axes>
```




```
loantodeposit = pd.DataFrame()

loantodeposit['yes'] = newdataframe[newdataframe['y'] == 'yes']['loan'].value_counts()
loantodeposit['no'] = newdataframe[newdataframe['y'] == 'no']['loan'].value_counts()

loantodeposit.sort_index().plot.bar(title = "Loan VS Deposit", figsize=(30,10))
```

```
<Axes: title={'center': 'Loan VS Deposit'}, xlabel='loan'>
```




```
old_age = newdataframe[newdataframe['age'] > 61]
```

```
old_age
```

```
      Unnamed: 0   age           job  marital  education default  balance housing loan   contact  day  month  duration  campaign  pdays  previous poutcome    y
29506       29507   66    housemaid  married  secondary      no     1929      no   no  cellular    2    feb       169         1     -1         0    unknown   no
29510       29511   62   technician  married  secondary      no      973      no   no  cellular    2    feb       326         1     -1         0    unknown  yes
29940       29941   62      retired  married  secondary      no     1495      no   no  cellular    2    feb       265         1     -1         0    unknown  yes
29158       29159   83      retired  married    primary      no      425      no   no  telephone   2    feb       912         1     -1         0    unknown   no
29241       29242   75      retired  divorced   primary      no       46      no   no  cellular    2    feb       294         1     -1         0    unknown   no
...           ...  ...          ...      ...        ...     ...      ...     ...  ...        ...  ...    ...       ...       ...    ...       ...        ...  ...
45191       45192   75      retired  divorced   tertiary     no     3810     yes   no  cellular   16    nov       262         1     193         1    failure  yes
45781       45782   68      retired  married  secondary      no     1146      no   no  cellular   16    nov       212         1     187         6    success  yes
45204       45205   73      retired  married  secondary      no     3092      no   no  cellular   17    nov       300         1      40         8    failure  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

[1021 rows x 18 columns]
```

```
old_age[old_age['y'] == 'yes']
```

```
      Unnamed: 0   age           job  marital  education default  balance housing loan   contact  day  month  duration  campaign  pdays  previous poutcome    y
29506       29507   66    housemaid  married  secondary      no     1929      no   no  cellular    2    feb       169         1     -1         0    unknown   no
29510       29511   62   technician  married  secondary      no      973      no   no  cellular    2    feb       326         1     -1         0    unknown  yes
29940       29941   62      retired  married  secondary      no     1495      no   no  cellular    2    feb       265         1     -1         0    unknown  yes
29158       29159   83      retired  married    primary      no      425      no   no  telephone   2    feb       912         1     -1         0    unknown   no
29241       29242   75      retired  divorced   primary      no       46      no   no  cellular    2    feb       294         1     -1         0    unknown   no
...           ...  ...          ...      ...        ...     ...      ...     ...  ...        ...  ...    ...       ...       ...    ...       ...        ...  ...
45191       45192   75      retired  divorced   tertiary     no     3810     yes   no  cellular   16    nov       262         1     193         1    failure  yes
45781       45782   68      retired  married  secondary      no     1146      no   no  cellular   16    nov       212         1     187         6    success  yes
45204       45205   73      retired  married  secondary      no     3092      no   no  cellular   17    nov       300         1      40         8    failure  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

[485 rows x 18 columns]
```

```
old_age
```

```
      Unnamed: 0   age           job  marital  education default  balance housing loan   contact  day  month  duration  campaign  pdays  previous poutcome    y
29506       29507   66    housemaid  married  secondary      no     1929      no   no  cellular    2    feb       169         1     -1         0    unknown   no
29510       29511   62   technician  married  secondary      no      973      no   no  cellular    2    feb       326         1     -1         0    unknown  yes
29940       29941   62      retired  married  secondary      no     1495      no   no  cellular    2    feb       265         1     -1         0    unknown  yes
29158       29159   83      retired  married    primary      no      425      no   no  telephone   2    feb       912         1     -1         0    unknown   no
29241       29242   75      retired  divorced   primary      no       46      no   no  cellular    2    feb       294         1     -1         0    unknown   no
...           ...  ...          ...      ...        ...     ...      ...     ...  ...        ...  ...    ...       ...       ...    ...       ...        ...  ...
45191       45192   75      retired  divorced   tertiary     no     3810     yes   no  cellular   16    nov       262         1     193         1    failure  yes
45781       45782   68      retired  married  secondary      no     1146      no   no  cellular   16    nov       212         1     187         6    success  yes
45204       45205   73      retired  married  secondary      no     3092      no   no  cellular   17    nov       300         1      40         8    failure  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

[1021 rows x 18 columns]
```

```
import matplotlib.pyplot as plt

yes_old_age = old_age[old_age['y'] == 'yes']

yes_old_age['age'].value_counts().plot.pie(autopct='%1.0f%%', startangle=90, figsize=(30, 30))
plt.title('Old Age Distribution for Yes Deposits')
plt.ylabel('')
plt.show()
```



```
newdataframe
```

```
      Unnamed: 0  age           job   marital  education default  balance housing loan   contact  day  month  duration  campaign  pdays  previous poutcome    y
0              0   58    management   married   tertiary      no     2143     yes   no  unknown    5    may       261         1     -1         0    unknown   no
1              1   44    technician    single  secondary      no       29     yes   no  unknown    5    may       151         1     -1         0    unknown   no
2              2   33  entrepreneur   married  secondary      no        2     yes  yes  unknown    5    may        76         1     -1         0    unknown   no
3              3   47   blue-collar   married        NaN      no     1506     yes   no  unknown    5    may        92         1     -1         0    unknown   no
4              5   33           NaN    single        NaN      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]
```

```
newdataframe['default'] = newdataframe['default'].replace({'yes': 1, 'no': 0})
```

```
/tmp/ipykernel_86/1655799637.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  newdataframe['default'] = newdataframe['default'].replace({'yes': 1, 'no': 0})
```

```
newdataframe['deposited'] = newdataframe['deposited'].replace({'yes': 1, 'no': 0})
```

```
/tmp/ipykernel_86/4240537748.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  newdataframe['deposited'] = newdataframe['deposited'].replace({'yes': 1, 'no': 0})
```

```
newdataframe['housing'] = newdataframe['housing'].replace({'yes': 1, 'no': 0})
```

```
/tmp/ipykernel_86/1307717748.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  newdataframe['housing'] = newdataframe['housing'].replace({'yes': 1, 'no': 0})
```

```
newdataframe['loan'] = newdataframe['loan'].replace({'yes': 1, 'no': 0})
```

```
/tmp/ipykernel_86/4161051918.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  newdataframe['loan'] = newdataframe['loan'].replace({'yes': 1, 'no': 0})
```

```
newdataframe
```

```
      Unnamed: 0  age           job   marital  education  default  balance  housing  loan   contact  day  month  duration  campaign  pdays  previous poutcome  deposited
0              0   58    management   married   tertiary        0     2143        1     0   unknown    5    may       261         1     -1         0  unknown          0
1              1   44    technician    single  secondary        0       29        1     0   unknown    5    may       151         1     -1         0  unknown          0
2              2   33  entrepreneur   married  secondary        0        2        1     1   unknown    5    may        76         1     -1         0  unknown          0
3              3   47   blue-collar   married        NaN        0     1506        1     0   unknown    5    may        92         1     -1         0  unknown          0
4              5   33           NaN    single        NaN        0        1        0     0   unknown    5    may       198         1     -1         0  unknown          0
...          ...  ...           ...       ...        ...      ...      ...      ...   ...       ...  ...    ...       ...       ...    ...       ...      ...        ...
45206      45207   51    technician   married   tertiary        0      825        0     0   cellular   17    nov       977         3     -1         0  unknown          1
45207      45208   71       retired  divorced    primary        0     1729        0     0   cellular   17    nov       456         2     -1         0  unknown          1
45208      45209   72       retired   married  secondary        0     5715        0     0   cellular   17    nov      1127         5     184         3  success          1
45209      45210   57   blue-collar   married  secondary        0      668        0     0  telephone   17    nov       508         4     -1         0  unknown          0
45210      45211   37  entrepreneur   married  secondary        0     2971        0     0   cellular   17    nov       361         2     188        11    other          0

[45211 rows x 18 columns]
```

```
newdataframe.dtypes
```

```
Unnamed: 0      int64
age             int64
job            object
marital        object
education      object
default         int64
balance         int64
housing         int64
loan            int64
contact        object
day             int64
month          object
duration        int64
campaign        int64
pdays           int64
previous        int64
poutcome       object
deposited       int64
dtype: object
```

```
newdataframe['default'] = newdataframe['default'].astype('int64')
```

```
newdataframe['housing'] = newdataframe['housing'].astype('int64')
```

```
newdataframe['loan'] = newdataframe['loan'].astype('int64')
```

```
newdataframe
```

```
      Unnamed: 0  age           job   marital  education  default  balance  housing  loan   contact  day  month  duration  campaign  pdays  previous poutcome  deposited
0              0   58    management   married   tertiary        0     2143        1     0   unknown    5    may       261         1     -1         0  unknown          0
1              1   44    technician    single  secondary        0       29        1     0   unknown    5    may       151         1     -1         0  unknown          0
2              2   33  entrepreneur   married  secondary        0        2        1     1   unknown    5    may        76         1     -1         0  unknown          0
3              3   47   blue-collar   married        NaN        0     1506        1     0   unknown    5    may        92         1     -1         0  unknown          0
4              5   33           NaN    single        NaN        0        1        0     0   unknown    5    may       198         1     -1         0  unknown          0
...          ...  ...           ...       ...        ...      ...      ...      ...   ...       ...  ...    ...       ...       ...    ...       ...      ...        ...
45206      45207   51    technician   married   tertiary        0      825        0     0   cellular   17    nov       977         3     -1         0  unknown          1
45207      45208   71       retired  divorced    primary        0     1729        0     0   cellular   17    nov       456         2     -1         0  unknown          1
45208      45209   72       retired   married  secondary        0     5715        0     0   cellular   17    nov      1127         5     184         3  success          1
45209      45210   57   blue-collar   married  secondary        0      668        0     0  telephone   17    nov       508         4     -1         0  unknown          0
45210      45211   37  entrepreneur   married  secondary        0     2971        0     0   cellular   17    nov       361         2     188        11    other          0

[45211 rows x 18 columns]
```

```
newdataframe.boxplot(column='age')
```

```
<Axes: >
```



```
newdataframe.drop('duration', axis='columns', inplace = True)
```

```
newdataframe
```

```
      Unnamed: 0  age           job   marital  education  default  balance  housing  loan   contact  day  month  campaign  pdays  previous poutcome  deposited
0              0   58    management   married   tertiary        0     2143        1     0   unknown    5    may         1     -1         0  unknown          0
1              1   44    technician    single  secondary        0       29        1     0   unknown    5    may         1     -1         0  unknown          0
2              2   33  entrepreneur   married  secondary        0        2        1     1   unknown    5    may         1     -1         0  unknown          0
3              3   47   blue-collar   married        NaN        0     1506        1     0   unknown    5    may         1     -1         0  unknown          0
4              5   33           NaN    single        NaN        0        1        0     0   unknown    5    may         1     -1         0  unknown          0
...          ...  ...           ...       ...        ...      ...      ...      ...   ...       ...  ...    ...       ...    ...       ...      ...        ...
45206      45207   51    technician   married   tertiary        0      825        0     0   cellular   17    nov         3     -1         0  unknown          1
45207      45208   71       retired  divorced    primary        0     1729        0     0   cellular   17    nov         2     -1         0  unknown          1
45208      45209   72       retired   married  secondary        0     5715        0     0   cellular   17    nov         5    184         3  success          1
45209      45210   57   blue-collar   married  secondary        0      668        0     0  telephone   17    nov         4     -1         0  unknown          0
45210      45211   37  entrepreneur   married  secondary        0     2971        0     0   cellular   17    nov         2    188        11    other          0

[45211 rows x 17 columns]
```

```
newdataframe.isna().sum()
```

```
Unnamed: 0    0
age           0
job           0
marital       0
education     0
default       0
balance       0
housing       0
loan          0
contact       0
day           0
month         0
campaign      0
pdays         0
previous      0
poutcome      0
deposited     0
dtype: int64
```

```
education    1857
default         0
housing         0
loan            0
contact         0
day             0
month           0
campaign        0
pdays           0
previous        0
poutcome        0
deposit         0
dtype: int64
```

```
newdataframe.shape
```

```
(45211, 17)
```

```
newdataframe = newdataframe.fillna({'default': 0, 'housing': 0, 'loan': 0})
```

```
newdataframe.isnull().sum()
```

```
age          0
job        288
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
campaign     0
pdays        0
previous     0
poutcome     0
deposit      0
dtype: int64
```

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

```
newdataframe.isnull().sum()
```

```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
campaign     0
pdays        0
previous     0
poutcome     0
deposit      0
dtype: int64
```

Double-click (or enter) to edit

Based on the analysis with the dataset for the Bank Marketing Campaign, several key business recommendations can be formulated to enhance the bank's marketing strategy for term purchases

(i) It is strongly recommended that the bank could directs its marketing efforts towards married couples. The Analysis chart indicates that the married individuals have demonstrated a notably higher chances of purchasing in term deposits.

(ii) The target demographic have to be considered as primary focus for the bank's promotional activities.

(iii) The bank can consider tailoring its marketing Campaign towards tertiary graduated individuals.This demographic has shown a higher propensity for term purchases, suggesting that targeting this educated group may yield higher chances of conversions.

(iv) Individuals without outstanding loans should be a priority in the bank's marketing strategy.

(v) The bank can concentrate its marketing endeavors on individuals in their late twenties to early forties.

(vi) The bank can concentrate on the customers with housing availablity for their term deposit product promotion

The demographic satisfying the classes can be identified to execute Bank Marketing Campaign as demonstrated in the above Exploratory Data Analysis.