

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

In [2]: bank_data = pd.read_csv('C:/Users/Eng Farhad/Desktop/Examp/Bank/Bank.csv')

In [3]: bank_data

Out[3]:
   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  unknown  no  1506  yes  no  unknown  5  may  92  1  -1  0  unknown  no
4  4  4  33  unknown  single  unknown  no  1  no  no  unknown  5  may  198  1  -1  0  unknown  no
...
45206  45206  51  technician  married  tertiary  no  825  no  no  cellular  17  nov  977  3  -1  0  unknown  yes
45207  45207  71  retired  divorced  primary  no  1729  no  no  cellular  17  nov  456  2  -1  0  unknown  yes
45208  45208  72  retired  married  secondary  no  5715  no  no  cellular  17  nov  1127  5  184  3  success  yes
45209  45209  57  blue-collar  married  secondary  no  668  no  no  telephone  17  nov  508  4  -1  0  unknown  no
45210  45210  37  entrepreneur  married  secondary  no  2971  no  no  cellular  17  nov  361  2  188  11  other  no
45211 rows x 18 columns

In [4]: #over review of columns, datatype and total records
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  Unnamed: 0  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(8), object(10)
memory usage: 6.2+ MB

In [5]: #summary of the data
bank_data.describe()

Out[5]:
   Unnamed: 0  age  balance  day  duration  campaign  pdays  previous
count  45211.000000  45211.000000  45211.000000  45211.000000  45211.000000  45211.000000  45211.000000
mean  22605.000000  40.936210  1362.272058  15.805419  258.163090  2.763941  40.197828  0.580323
std  13051.435847  10.618762  3044.766829  8.322476  257.527812  3.098021  100.128746  2.303441
min  0.000000  18.000000  -8019.000000  1.000000  0.000000  1.000000  -1.000000  0.000000
25%  11302.500000  33.000000  72.000000  8.000000  103.000000  1.000000  -1.000000  0.000000
50%  22605.000000  39.000000  448.000000  16.000000  180.000000  2.000000  -1.000000  0.000000
75%  33907.500000  48.000000  1428.000000  21.000000  319.000000  3.000000  -1.000000  0.000000
max  45210.000000  95.000000  102127.000000  31.000000  4918.000000  63.000000  871.000000  275.000000

In [7]: #shape of (rows and columns) of the dataset
bank_data.shape

Out[7]: (45211, 18)

In [9]: # the y column actually mean if a customer converted to easily manipulate data we need to change yes and no to 0 and 1
bank_data['converted'] = bank_data['y'].apply(lambda x: 0 if x=='no' else 1)

In [11]: bank_data['converted'].unique()

Out[11]: array([0, 1], dtype=int64)

In [12]: del bank_data['y']

In [13]: bank_data.head()

Out[13]:
   Unnamed: 0  age  job  marital  education  default  balance  housing  loan  contact  day  month  duration  campaign  pdays  previous  poutcome  converted
0  0  58  management  married  tertiary  no  2143  yes  no  unknown  5  may  261  1  -1  0  unknown  0
1  1  44  technician  single  secondary  no  29  yes  no  unknown  5  may  151  1  -1  0  unknown  0
2  2  33  entrepreneur  married  secondary  no  2  yes  yes  unknown  5  may  76  1  -1  0  unknown  0
3  3  47  blue-collar  married  unknown  no  1506  yes  no  unknown  5  may  92  1  -1  0  unknown  0
4  4  4  33  unknown  single  unknown  no  1  no  no  unknown  5  may  198  1  -1  0  unknown  0

In [27]: # converted and balance boxplot with outliers
ax = bank_data[['converted', 'balance']].boxplot(bys='converted', showfliers=True, figsize=(7,5))
ax.set_ylabel('Average of Bank Balance')
ax.set_xlabel('Conversion')
ax.set_title('Average Bank Balance and Conversion')
plt.suptitle('')
plt.show()

In [34]: #converted and balance without outliers
ax = bank_data[['converted', 'balance']].boxplot(bys='converted', showfliers=False, figsize=(9,5))
ax.set_xlabel('Converted')
ax.set_ylabel('Average Bank Balance')
ax.set_title('Average Bank Balance by conversion (0 = no conversion and 1 = conversion)')
plt.suptitle('')
plt.show()

In [44]: #lets visualize converted and bank balance using violinplot
fig, axes = plt.subplots(figsize=(10,6))
sns.violinplot('converted', 'balance', data=bank_data, axis=axes)
axes.set_xlabel('Converted')
axes.set_ylabel('Average Bank Balance')
axes.set_title('Bank Balance and Conversion')

C:\Users\Eng Farhad\anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be data, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Out[44]: Text(0.5, 1.0, 'Bank Balance and Conversion')

In [64]: percentage_campaign = bank_data.groupby('campaign')[['converted']].sum() / bank_data.groupby('campaign')[['converted']].count()*100
#top 10 campaign which had positive result onn converting customer
percentage_campaign.head(10)

Out[64]:
campaign
14  0.997583
11  2.095319
1  1.193624
9  0.809068
7  7.879819
6  7.126259
7  6.394558
8  5.925926
9  6.422018
10  5.283158
Name: converted, dtype: float64

In [74]: #conversion plot by campaign
percentage_campaign.plot(kind='bar', figsize=(15,6), color='coral', grid=True)
plt.xlabel('Campaign')
plt.ylabel('Percentage % of Conversion')
plt.title('Percentage of conversion over Number of Campaign contacts', fontweight='bold')
plt.show()

In [72]: bank_data.head()

Out[72]:
   Unnamed: 0  age  job  marital  education  default  balance  housing  loan  contact  day  month  duration  campaign  pdays  previous  poutcome  converted
0  0  58  management  married  tertiary  no  2143  yes  no  unknown  5  may  261  1  -1  0  unknown  0
1  1  44  technician  single  secondary  no  29  yes  no  unknown  5  may  151  1  -1  0  unknown  0
2  2  33  entrepreneur  married  secondary  no  2  yes  yes  unknown  5  may  76  1  -1  0  unknown  0
3  3  47  blue-collar  married  unknown  no  1506  yes  no  unknown  5  may  92  1  -1  0  unknown  0
4  4  4  33  unknown  single  unknown  no  1  no  no  unknown  5  may  198  1  -1  0  unknown  0

In [86]: #number of customer converted compare to thier job
conversion_by_job = bank_data.groupby('job')['converted'].sum().plot(kind='bar', figsize=(12,6), color='coral')
plt.xlabel('Job Category')
plt.ylabel('Number of converted Customer')
plt.title('Number of converted Custober by Job')
plt.xticks(rotation=30)

Out[86]:
(array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]),
[Text(0, 0, 'admin.'),
Text(1, 0, 'blue-collar'),
Text(2, 0, 'entrepreneur'),
Text(3, 0, 'housemaid'),
Text(4, 0, 'management'),
Text(5, 0, 'retired'),
Text(6, 0, 'self-employed'),
Text(7, 0, 'services'),
Text(8, 0, 'student'),
Text(9, 0, 'technician'),
Text(10, 0, 'unemployed'),
Text(11, 0, 'unknown')])

In [101]: # Percentage of conversion compare to customer designation
conversion_by_job_per = round(bank_data.groupby('job')['converted'].sum() / bank_data.groupby('job')['converted'].count()*100, 2)

In [102]: conversion_by_job_per

Out[102]:
job
admin. 12.29
blue-collar 7.27
entrepreneur 8.27
housemaid 8.79
management 13.76
retired 22.79
self-employed 11.84
services 8.88
student 28.68
technician 11.88
unemployed 15.50
unknown 11.81
Name: converted, dtype: float64

In [108]: #job conversion percentage
conversion_by_job_per.plot(kind='bar', figsize=(12, 6), color='blue')
plt.xlabel('Job Category')
plt.ylabel('Conversion Percentage %')
plt.title('Percentage % of conversion over Job Category', fontweight='bold', fontsize=14)
plt.xticks(rotation=30, fontweight='bold')

Out[108]:
(array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]),
[Text(0, 0, 'admin.'),
Text(1, 0, 'blue-collar'),
Text(2, 0, 'entrepreneur'),
Text(3, 0, 'housemaid'),
Text(4, 0, 'management'),
Text(5, 0, 'retired'),
Text(6, 0, 'self-employed'),
Text(7, 0, 'services'),
Text(8, 0, 'student'),
Text(9, 0, 'technician'),
Text(10, 0, 'unemployed'),
Text(11, 0, 'unknown')])

In [110]: bank_data.head()

Out[110]:
   Unnamed: 0  age  job  marital  education  default  balance  housing  loan  contact  day  month  duration  campaign  pdays  previous  poutcome  converted
0  0  58  management  married  tertiary  no  2143  yes  no  unknown  5  may  261  1  -1  0  unknown  0
1  1  44  technician  single  secondary  no  29  yes  no  unknown  5  may  151  1  -1  0  unknown  0
2  2  33  entrepreneur  married  secondary  no  2  yes  yes  unknown  5  may  76  1  -1  0  unknown  0
3  3  47  blue-collar  married  unknown  no  1506  yes  no  unknown  5  may  92  1  -1  0  unknown  0
4  4  4  33  unknown  single  unknown  no  1  no  no  unknown  5  may  198  1  -1  0  unknown  0

In [113]: bank_data.columns

Out[113]: Index(['Unnamed: 0', 'age', 'job', 'marital', 'education', 'default', 'balance', 'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays', 'previous', 'poutcome', 'converted'],
      dtype='object')

In [125]: #groupby month
bank_data.groupby('month')['converted'].count()

Out[125]:
month
apr  2932
aug  6247
dec  214
feb  2649
jan  1489
jul  6895
jun  5341
mar  477
may  13766
nov  3970
oct  738
sep  978
Name: converted, dtype: int64

In [131]: #converting string months into number
month = bank_data['month'].unique()
months = (['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec'])
bank_data['month_in_num'] = bank_data['month'].apply(lambda x: months.index(x)+1)
bank_data['month_in_num']

Out[136]:
0  5
1  5
2  1
3  5
4  5
45206  11
45207  11
45208  11
45209  11
45210  11
Name: month_in_num, Length: 45211, dtype: int64

In [137]: bank_data['job'].unique()

Out[137]: array(['management', 'technician', 'entrepreneur', 'blue-collar', 'unknown', 'retired', 'admin.', 'services', 'self-employed', 'unemployed', 'housemaid', 'student'], dtype=object)

In [138]: bank_data.head(3)

Out[138]:
   Unnamed: 0  age  job  marital  education  default  balance  housing  loan  contact  day  month  duration  campaign  pdays  previous  poutcome  converted  month_in_num
0  0  58  management  married  tertiary  no  2143  yes  no  unknown  5  may  261  1  -1  0  unknown  0  5
1  1  44  technician  single  secondary  no  29  yes  no  unknown  5  may  151  1  -1  0  unknown  0  5
2  2  33  entrepreneur  married  secondary  no  2  yes  yes  unknown  5  may  76  1  -1  0  unknown  0  5

In [139]: bank_data['marital'].unique()

Out[139]: array(['married', 'single', 'divorced'], dtype=object)

In [140]: bank_data['housing'].unique()

Out[140]: array(['yes', 'no'], dtype=object)

In [141]: #replacing housing from yes and no into 1 and 0
bank_data['housing'] = bank_data['housing'].apply(lambda x: 0 if x=='no' else 1)

In [143]: bank_data['housing'].head(5)

Out[143]:
0  1
1  1
2  1
3  1
4  0
Name: housing, dtype: int64

In [144]: bank_data['loan'].unique()

Out[144]: array(['no', 'yes'], dtype=object)

In [145]: #converting loan column into numeric as its yes and no
bank_data['loan'] = bank_data['loan'].apply(lambda x: 0 if x=='no' else 1)

In [147]: bank_data['loan'].head(5)

Out[147]:
0  0
1  0
2  1
3  0
4  0
Name: loan, dtype: int64

In [148]: bank_data.head(5)

Out[148]:
   Unnamed: 0  age  job  marital  education  default  balance  housing  loan  contact  day  month  duration  campaign  pdays  previous  poutcome  converted  month_in_num
0  0  58  management  married  tertiary  no  2143  yes  no  unknown  5  may  261  1  -1  0  unknown  0  5
1  1  44  technician  single  secondary  no  29  yes  no  unknown  5  may  151  1  -1  0  unknown  0  5
2  2  33  entrepreneur  married  secondary  no  2  yes  yes  unknown  5  may  76  1  -1  0  unknown  0  5
3  3  47  blue-collar  married  unknown  no  1506  yes  no  unknown  5  may  92  1  -1  0  unknown  0  5
4  4  4  33  unknown  single  unknown  no  1  no  no  unknown  5  may  198  1  -1  0  unknown  0  5

In [149]: #removing month column because we already converted into number month
del bank_data['month']

In [150]: bank_data['default'].unique()

Out[150]: array(['no', 'yes'], dtype=object)

In [151]: # meanwhile default column is also yes no string we are going to convert it into number 1 and 0
bank_data['default'] = bank_data['default'].apply(lambda x: 0 if x=='no' else 1)

In [152]: bank_data.head(3)

Out[152]:
   Unnamed: 0  age  job  marital  education  default  balance  housing  loan  contact  day  duration  campaign  pdays  previous  poutcome  converted  month_in_num
0  0  58  management  married  tertiary  0  2143  yes  no  unknown  5  261  1  -1  0  unknown  0  5
1  1  44  technician  single  secondary  0  29  yes  no  unknown  5  151  1  -1  0  unknown  0  5
2  2  33  entrepreneur  married  secondary  0  2  yes  yes  unknown  5  76  1  -1  0  unknown  0  5

In [153]: bank_data['education'].unique()

Out[153]: array(['tertiary', 'secondary', 'unknown', 'primary'], dtype=object)

In [155]: #converting education column into sub columns as numeric
bank_data = pd.get_dummies(data=bank_data, columns=['education'])

In [156]: bank_data

Out[156]:
   Unnamed: 0  age  job  marital  default  balance  housing  loan  contact  day  duration  campaign  pdays  previous  poutcome  converted  month_in_num  education_primary  education_secor
0  0  58  management  married  0  2143  yes  no  unknown  5  ...  1  -1  0  unknown  0  5  0  0
1  1  44  technician  single  0  29  yes  no  unknown  5  ...  1  -1  0  unknown  0  5  0  0
2  2  33  entrepreneur  married  0  2  yes  yes  unknown  5  ...  1  -1  0  unknown  0  5  0  0
3  3  47  blue-collar  married  0  1506  yes  no  unknown  5  ...  1  -1  0  unknown  0  5  0  0
4  4  4  33  unknown  single  0  1  no  no  unknown  5  ...  1  -1  0  unknown  0  5  0  0
...
45206  45206  51  technician  divorced  0  825  no  no  cellular  17  ...  3  -1  0  unknown  1  11  0  0
45207  45207  71  retired  divorced  0  1729  no  no  cellular  17  ...  2  -1  0  unknown  1  11  1  0
45208  45208  72  retired  married  0  5715  no  no  cellular  17  ...  5  184  3  success  1  11  0  0
45209  45209  57  blue-collar  married  0  668  no  no  telephone  17  ...  4  -1  0  unknown  0  11  0  0
45210  45210  37  entrepreneur  married  0  2971  no  no  cellular  17  ...  2  188  11  other  0  11  0  0
45211 rows x 21 columns

In [157]: bank_data = pd.get_dummies(data=bank_data, columns=['job'])

In [158]: bank_data

Out[158]:
   Unnamed: 0  age  marital  default  balance  housing  loan  contact  day  duration  ...  job_entrepreneur  job_housemaid  job_management  job_retired  job_self-employed  job_services  job_student  job_
0  0  58  married  0  2143  yes  no  unknown  5  261  ...  0  0  0  0  0  0  0  0  0  0  0  0
1  1  44  single  0  29  yes  no  unknown  5  151  ...  0  0  0  0  0  0  0  0  0  0  0  0
2  2  33  married  0  2  yes  yes  unknown  5  76  ...  1  0  0  0  0  0  0  0  0  0  0  0
3  3  47  married  0  1506  yes  no  unknown  5  92  ...  0  0  0  0  0  0  0  0  0  0  0  0
4  4  4  single  0  1  no  no  unknown  5  198  ...  0  0  0  0  0  0  0  0  0  0  0  0
...
45206  45206  51  no  825  no  no  cellular  17  977  ...  0  0  0  0  0  0  0  0  0  0  0  0
45207  45207  71  divorced  0  1729  no  no  cellular  17  456  ...  0  0  0  0  0  1  0  0  0  0  0  0
45208  45208  72  married  0  5715  no  no  cellular  17  1127  ...  0  0  0  0  0  0  0  0  0  0  0  0
45209  45209  57  married  0  668  no  no  telephone  17  508  ...  0  0  0  0  0  0  0  0  0  0  0  0
45210  45210  37  married  0  2971  no  no  cellular  17  361  ...  1  0  0  0  0  0  0  0  0  0  0  0
45211 rows x 32 columns

In [159]: bank_data = pd.get_dummies(data=bank_data, columns=['marital'])

In [160]: bank_data

Out[160]:
   Unnamed: 0  age  default  balance  housing  loan  contact  day  duration  campaign  ...  job_retired  job_self-employed  job_services  job_student  job_technician  job_unemployed  job_unknown  marital
0  0  58  0  2143  yes  no  unknown  5  261  1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
1  1  44  0  29  yes  no  unknown  5  151  1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
2  2  33  0  2  yes  yes  unknown  5  76  1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
3  3  47  0  1506  yes  no  unknown  5  92  1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
4  4  4  33  0  1  no  no  unknown  5  198  1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
...
45206  45206  51  0  825  no  no  cellular  17  977  3  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
45207  45207  71  0  1729  no  no  cellular  17  456  2  ...  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45208  45208  72  0  5715  no  no  cellular  17  1127  5  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45209  45209  57  0  668  no  no  telephone  17  508  4  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45210  45210  37  0  2971  no  no  cellular  17  361  2  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45211 rows x 34 columns

In [161]: bank_data['contact'].unique()

Out[161]: array(['unknown', 'cellular', 'telephone'], dtype=object)

In [162]: bank_data = pd.get_dummies(data=bank_data, columns=['contact'])

In [163]: bank_data

Out[163]:
   Unnamed: 0  age  default  balance  housing  loan  day  duration  campaign  pdays  ...  job_student  job_technician  job_unemployed  marital_divorced  marital_married  marital_single  contact_cellular  contact_telephone  contact_unknown  pos
0  0  58  0  2143  yes  no  5  261  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
1  1  44  0  29  yes  no  5  151  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
2  2  33  0  2  yes  yes  5  76  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
3  3  47  0  1506  yes  no  5  92  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
4  4  4  33  0  1  no  no  5  198  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
...
45206  45206  51  0  825  no  no  17  977  3  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
45207  45207  71  0  1729  no  no  17  456  2  -1  ...  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45208  45208  72  0  5715  no  no  17  1127  5  184  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45209  45209  57  0  668  no  no  17  508  4  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45210  45210  37  0  2971  no  no  17  361  2  188  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45211 rows x 39 columns

In [165]: bank_data['poutcome'].unique()

Out[165]: array(['unknown', 'failure', 'other', 'success'], dtype=object)

In [166]: bank_data = pd.get_dummies(data=bank_data, columns=['poutcome'])

In [167]: bank_data

Out[167]:
   Unnamed: 0  age  default  balance  housing  loan  day  duration  campaign  pdays  ...  marital_divorced  marital_married  marital_single  contact_cellular  contact_telephone  contact_unknown  pos
0  0  58  0  2143  yes  no  5  261  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
1  1  44  0  29  yes  no  5  151  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
2  2  33  0  2  yes  yes  5  76  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
3  3  47  0  1506  yes  no  5  92  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
4  4  4  33  0  1  no  no  5  198  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
...
45206  45206  51  0  825  no  no  17  977  3  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
45207  45207  71  0  1729  no  no  17  456  2  -1  ...  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45208  45208  72  0  5715  no  no  17  1127  5  184  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45209  45209  57  0  668  no  no  17  508  4  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45210  45210  37  0  2971  no  no  17  361  2  188  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45211 rows x 39 columns

In [168]: bank_data.get_numeric_data()

Out[168]:
   Unnamed: 0  age  default  balance  housing  loan  day  duration  campaign  pdays  ...  marital_divorced  marital_married  marital_single  contact_cellular  contact_telephone  contact_unknown  pos
0  0  58  0  2143  yes  no  5  261  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
1  1  44  0  29  yes  no  5  151  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
2  2  33  0  2  yes  yes  5  76  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
3  3  47  0  1506  yes  no  5  92  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
4  4  4  33  0  1  no  no  5  198  1  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
...
45206  45206  51  0  825  no  no  17  977  3  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
45207  45207  71  0  1729  no  no  17  456  2  -1  ...  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45208  45208  72  0  5715  no  no  17  1127  5  184  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45209  45209  57  0  668  no  no  17  508  4  -1  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45210  45210  37  0  2971  no  no  17  361  2  188  ...  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
45211 rows x 39 columns

In [169]: #now we have converted all categorical data into numeric and its ready to implement machine learning concepts.
```