Data Science Internship

Week 8: Data Science Project: Bank Marketing (Campaign)

Group Name: One
Name: Hira Fahim
Email: hirashahidd26@yahoo.com
Country: United Kingdom
Company: Unemployed
Specialization: Data Science
Batch Code: LISUM19
Submission Date: 8th April 2023
Submitted to: Data Glacier

Table of Contents

1. Problem Description

ABC Bank wants to sell its 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).

2. Dataset 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, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

3. Type of the Data used for Analysis.

3.1. Bank client data

The data contains details related to the clients' age, job type, marital status, education level, credit history (default or not), housing loan and personal loans.

3.2. Data related to the last contact of the current campaign.

This part of data contains information related to the way of contact (cellular, mobile etc) and information about time duration of contact like how many days have passed since they contact the client and on which month, they have contacted the client and total call duration in seconds.

3.3. Other Attributes

These attributes are related to the campaign and clients' contact since previous campaign. It includes input variables like campaign, pdays (number of days that passed by after the client was last contacted from a previous campaign), previous (number of contacts performed before this campaign and for this client) and poutcome (outcome of the previous marketing campaign).

4. Attribute Information

4.1. Input Variables

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: 'primary', 'secondary', 'tertiary', 'unknown')
5. default: has credit in default? (Categorical: 'no', 'yes')
6. balance: average yearly balance, in euros (numerical)
7. housing: has housing loan? (Categorical: 'no', 'yes')
8. loan: has personal loan? (Categorical: 'no', 'yes')
9. contact: contact communication type (categorical: 'cellular', 'telephone', Unknown)
10. day: last contact day of the month (numeric)
11. month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
12. duration: last contact duration, in seconds (numerical)
13. campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14. 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)
15. previous: number of contacts performed before this campaign and for this client (numeric)
16. poutcome: outcome of the previous marketing campaign (categorical: 'failure', ‘unknown', 'success', ‘other’)

4.2. Output variable (desired target)

17. y - has the client subscribed a term deposit? (Binary: 'yes', 'no')

5. Import Data Set

```
In [1]: import pandas as pd
        import numpy as np

In [2]: d= pd.read_csv("bank-full.csv")

In [3]: d.head()
Out[3]:
```

	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

6. Dataset Details

- Shape of the dataset (Number of rows and columns)

```
In [18]: # no of rows and columns
         d.shape

Out[18]: (45211, 17)
```

Number of rows = 45211
Number of columns = 17

- Datatype of Columns and Non-null values

```
In [19]: # Datatypes of columns and non-null values
         d.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
```

➢ Numerical and categorical Features

```
Numeric Features:
Index(['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous'], dtype='object')
================================================================================
Categorical Features:
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome', 'y'],
      dtype='object')
```

7. Problems in the data

7.1. Null values

```
# total null values in the dataset
d.isnull().sum()

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

There are no null values in the dataset.

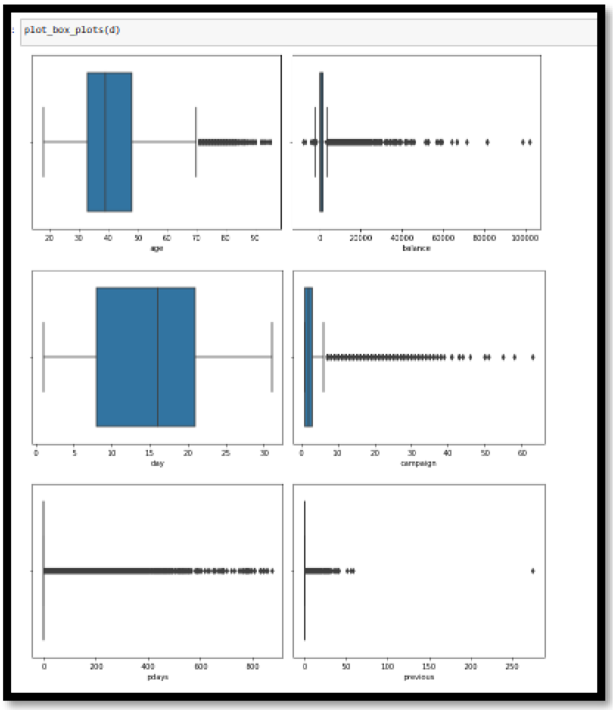
7.2. Outliers and skewness

➢ Description of Numerical column

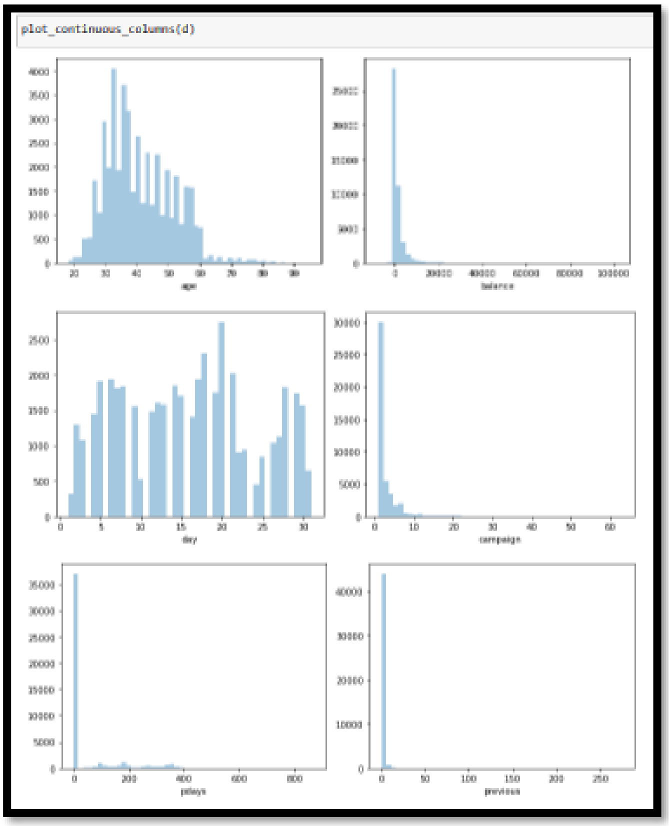
```
# Description of numerical columns
d.describe()
```

	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

- Numerical variables' visualization

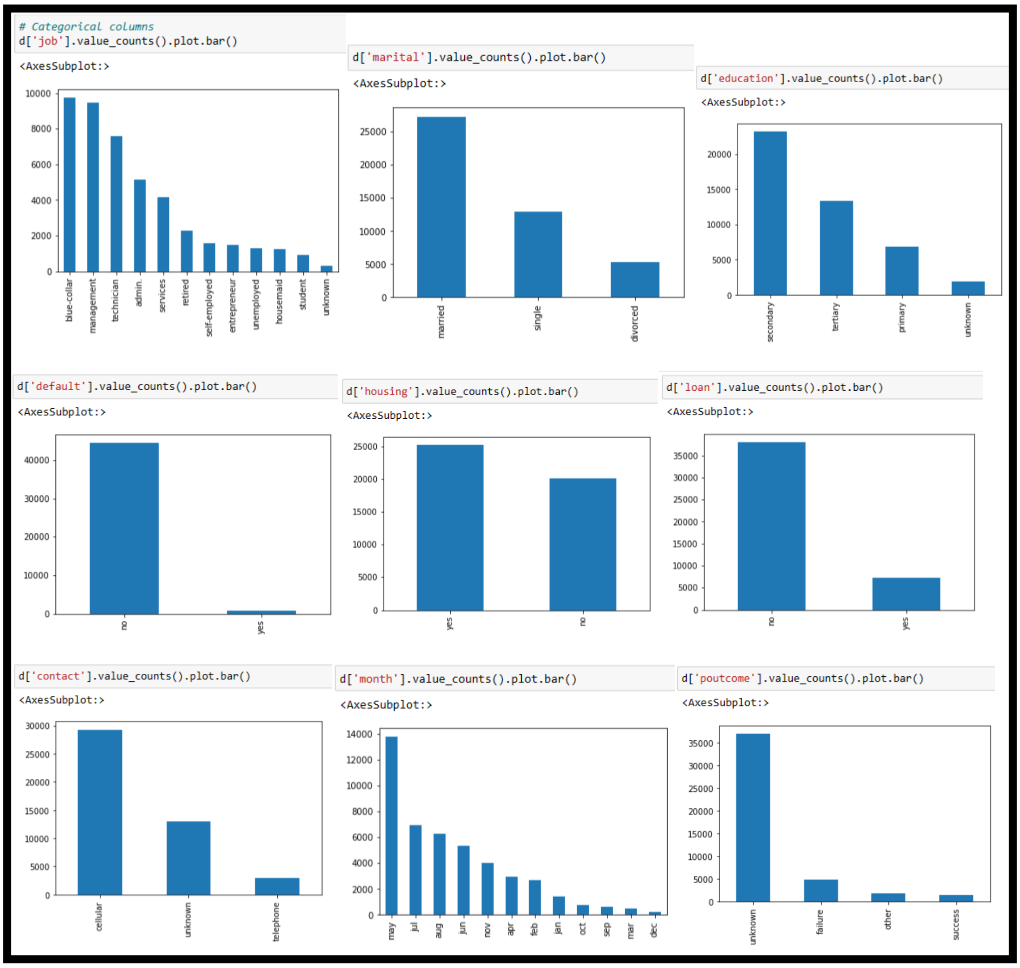


From **description** and **boxplot**, we can see there are outliers in numerical input variables like age, balance, campaign, pdays and previous.



In Histogram, we can see input variables like age, balance, campaign, pdays and previous are **positively skewed**, and we can also se uneven distribution of data in day column.

➢ Categorical data visualization



In **Bar chart** of categorical columns, we see uneven distribution of data in almost all the input categorical columns.

8. Feature Scaling -Normalization

To deal with noises in the data, we need to perform feature scaling and as there are both continuous and discrete columns, we are using **normalization scaling technique** to transform features to be on a similar scale. This improves the performance and training stability of the model.

```
In [67]: from numpy import set_printoptions
         from sklearn.preprocessing import MinMaxScaler

In [68]: d1=d.iloc[:,:-1]
         d1
```

Out[68]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	campaign	pdays	previous	poutcome
0	58	4	1	2	0	2143	1	0	2	5	8	1	-1	0	3
1	44	9	2	1	0	29	1	0	2	5	8	1	-1	0	3
2	33	2	1	1	0	2	1	1	2	5	8	1	-1	0	3
3	47	1	1	3	0	1506	1	0	2	5	8	1	-1	0	3
4	33	11	2	3	0	1	0	0	2	5	8	1	-1	0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	9	1	2	0	825	0	0	0	17	9	3	-1	0	3
45207	71	5	0	0	0	1729	0	0	0	17	9	2	-1	0	3
45208	72	5	1	1	0	5715	0	0	0	17	9	5	184	3	2
45209	57	1	1	1	0	668	0	0	1	17	9	4	-1	0	3
45210	37	2	1	1	0	2971	0	0	0	17	9	2	188	11	1

45211 rows × 15 columns

In [69]:

```
array=d1.values
scaler=MinMaxScaler(feature_range=(0,1))
rescaledX=scaler.fit_transform(array)

set_printoptions(precision=2)
print(rescaledX[0:5,:])
```

```
[[0.52 0.36 0.5  0.67 0.   0.09 1.   0.   1.   0.13 0.73 0.   0.   0.
  1.  ]
 [0.34 0.82 1.   0.33 0.   0.07 1.   0.   1.   0.13 0.73 0.   0.   0.
  1.  ]
 [0.19 0.18 0.5  0.33 0.   0.07 1.   1.   1.   0.13 0.73 0.   0.   0.
  1.  ]
 [0.38 0.09 0.5  1.   0.   0.09 1.   0.   1.   0.13 0.73 0.   0.   0.
  1.  ]
 [0.19 1.   1.   1.   0.   0.07 0.   0.   1.   0.13 0.73 0.   0.   0.
  1.  ]]
```

In [70]:

```
d2=pd.DataFrame(rescaledX,columns=["age","job","marital","education","default","balance","housing","loan","contact","day",
                                   "month","campaign","pdays","previous","poutcome"])
d2
```

Out[70]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	campaign	pdays	previous	poutcome
0	0.519481	0.363636	0.5	0.666667	0.0	0.092259	1.0	0.0	1.0	0.133333	0.727273	0.000000	0.000000	0.000000	1.000000
1	0.337662	0.818182	1.0	0.333333	0.0	0.073067	1.0	0.0	1.0	0.133333	0.727273	0.000000	0.000000	0.000000	1.000000
2	0.194805	0.181818	0.5	0.333333	0.0	0.072822	1.0	1.0	1.0	0.133333	0.727273	0.000000	0.000000	0.000000	1.000000
3	0.376623	0.090909	0.5	1.000000	0.0	0.086476	1.0	0.0	1.0	0.133333	0.727273	0.000000	0.000000	0.000000	1.000000
4	0.194805	1.000000	1.0	1.000000	0.0	0.072812	0.0	0.0	1.0	0.133333	0.727273	0.000000	0.000000	0.000000	1.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	0.428571	0.818182	0.5	0.666667	0.0	0.080293	0.0	0.0	0.0	0.533333	0.818182	0.032258	0.000000	0.000000	1.000000
45207	0.688312	0.454545	0.0	0.000000	0.0	0.088501	0.0	0.0	0.0	0.533333	0.818182	0.016129	0.000000	0.000000	1.000000
45208	0.701299	0.454545	0.5	0.333333	0.0	0.124689	0.0	0.0	0.0	0.533333	0.818182	0.064516	0.212156	0.010909	0.666667
45209	0.506494	0.090909	0.5	0.333333	0.0	0.078868	0.0	0.0	0.5	0.533333	0.818182	0.048387	0.000000	0.000000	1.000000
45210	0.246753	0.181818	0.5	0.333333	0.0	0.099777	0.0	0.0	0.0	0.533333	0.818182	0.016129	0.216743	0.040000	0.333333

45211 rows × 15 columns