Week 8

Data Science Intern at Data Glacier
Project: Bank Marketing (Campaign)

Name: Adrian Baysa
Email: adrianbaysa2@gmail.com
Country: Philippines
Batch Code: LISUM16

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. **Data Understanding**

   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). The bank-full.csv dataset has all examples and 17 inputs, ordered by date (older version of this dataset with less inputs).

3. **Data Types**

   The following are the details of the features and target variable:

   a. 1 - age (numeric)
   b. 2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
   c. 3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
   d. 4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
   e. 5 - default: has credit in default? (categorical: 'no','yes','unknown')
   f. 6 - housing: has housing loan? (categorical: 'no','yes','unknown')
   g. 7 - loan: has personal loan? (categorical: 'no','yes','unknown')
      # related with the last contact of the current campaign:
   h. 8 - contact: contact communication type (categorical: 'cellular','telephone')
   i. 9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
   j. 10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
   k. 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:

l. 12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
m. 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)
n. 14 - previous: number of contacts performed before this campaign and for this client (numeric)
o. 15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
   # social and economic context attributes
p. 16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
q. 17 - cons.price.idx: consumer price index - monthly indicator (numeric)
r. 18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
s. 19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
t. 20 - nr.employed: number of employees - quarterly indicator (numeric)
u. Output variable (desired target): 21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

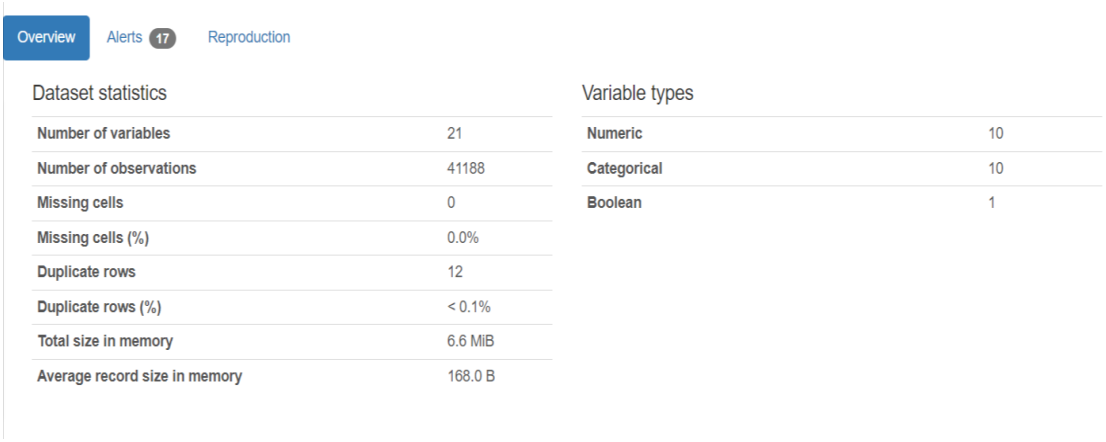
Further details of the data types and the sample of the dataframe are presented below:

4. Problems with the data

a. For the data set used, no missing data or duplicate rows were encountered. However, duplicates were encountered and were deleted.

```
[3]   # Check for missing values
      print('Data columns with null values:',df.isnull().sum(), sep = '\n')
```

```
Data columns with null values:
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
y                 0
dtype: int64
```



Overview | Alerts 17 | Reproduction

Dataset statistics

Number of variables	21
Number of observations	41188
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	12
Duplicate rows (%)	< 0.1%
Total size in memory	6.6 MiB
Average record size in memory	168.0 B

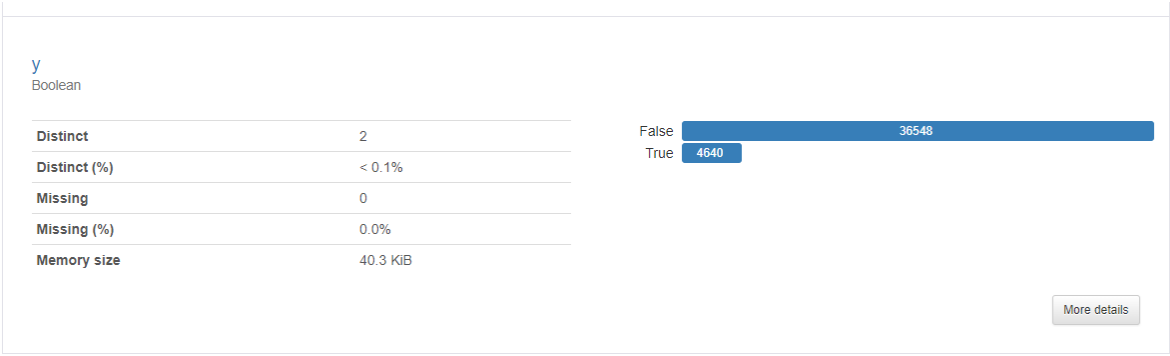
Variable types

Numeric	10
Categorical	10
Boolean	1

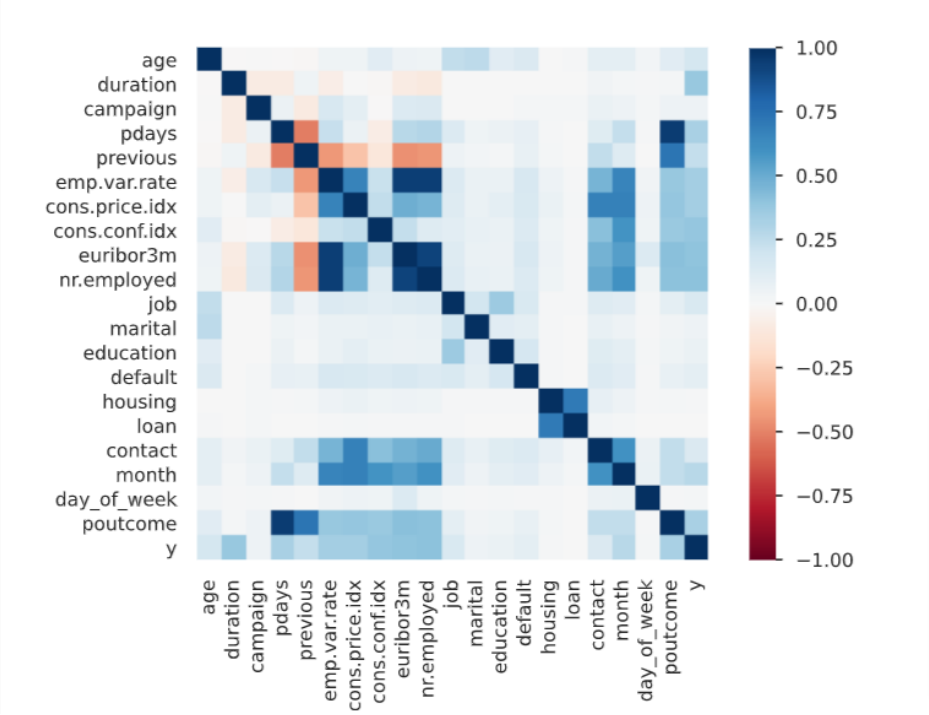
Data Cleansing and Features Engineering

```
[ ]   # Drop the duration column as required
      df.drop(['duration'], axis=1, inplace=True)
      df.info()

      # Drop duplicates
      duplicated_rows = df[df.duplicated()]
      print('Number of duplicated rows:', duplicated_rows.shape[0])

      df.drop_duplicates(inplace = True)
      print(df.shape)
```

The target variables (y) is also highly imbalanced and will need to be dealt with during the training and testing.



b. Some features are strongly correlated, but can easily be managed. For example, the euribor.3m and emp.var.rate are highly correlated.



Heatmap | Table

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	job	ma
age	1.000	-0.002	0.006	-0.001	-0.013	0.045	0.045	0.115	0.054	0.045	0.249	0.2
duration	-0.002	1.000	-0.081	-0.083	0.042	-0.069	0.003	-0.009	-0.078	-0.095	0.006	0.0
campaign	0.006	-0.081	1.000	0.056	-0.087	0.156	0.096	-0.002	0.141	0.144	0.000	0.0
pdays	-0.001	-0.083	0.056	1.000	-0.510	0.228	0.057	-0.077	0.278	0.291	0.140	0.0
previous	-0.013	0.042	-0.087	-0.510	1.000	-0.435	-0.283	-0.116	-0.455	-0.439	0.053	0.0
emp.var.rate	0.045	-0.069	0.156	0.228	-0.435	1.000	0.665	0.225	0.940	0.945	0.135	0.0
cons.price.idx	0.045	0.003	0.096	0.057	-0.283	0.665	1.000	0.246	0.491	0.465	0.132	0.0
cons.conf.idx	0.115	-0.009	-0.002	-0.077	-0.116	0.225	0.246	1.000	0.237	0.133	0.109	0.0
euribor3m	0.054	-0.078	0.141	0.278	-0.455	0.940	0.491	0.237	1.000	0.929	0.128	0.0

5. **Approaches**
    a. Impute missing values, if any
    b. Remove duplicates
    c. Imputation for outliers
    d. Bin columns (e.g. pdays into aging buckets, etc.)
    e. Do features engineering to reduce the cardinality of categorical data

6. **Github Repolink:** https://github.com/agbaysa/dataglacier_week8