Data Science Project

Bank Marketing(Campaign)

Group Name: Team Winning Mode

Team Members:

1. Pooja Honneshwari Ravi
- poojaravi385@gmail.com, USA,
California State University-Fullerton, Data Science

2. Guru Guha Narayanan Muthunarayanan
- nguruguha@gmail.com, UK,
University of Manchester, Data Science

3. Joyeta Saha
- mim.joye@gmail.com, UK,
University of Surrey, Data Science

4. Darshan Patil
- darshan.patil1128@gmail.com, USA,
Rutgers University, Data Science

Problem Description:

ABC Bank is planning to launch a new term deposit product and wants to identify potential customers who are more likely to purchase the product. They aim to develop a machine learning (ML) model based on customers' past interactions with the bank or other financial institutions. The purpose of this model is to assist in shortlisting customers with a higher probability of buying the term deposit, allowing the bank's marketing channels to focus their efforts and resources on those customers. By targeting the right customers, ABC Bank aims to optimize their marketing campaigns and save valuable resources and time.

Data Understanding:

The data is related to direct marketing initiatives of a Portuguese banking organization. Calls served as the foundation for the marketing campaigns. It was frequently necessary to make multiple contacts with the same client in order to determine if the product (bank term deposit) would be subscribed ('yes') or not ('no').

There are four datasets:

1. bank-additional-full.csv with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010), very close to the data analyzed in [Moro et al., 2014]
2. bank-additional.csv with 10% of the examples (4119), randomly selected from 1), and 20 inputs.
3. bank-full.csv with all examples and 17 inputs, ordered by date (older version of this dataset with less inputs).
4. bank.csv with 10% of the examples and 17 inputs, randomly selected from 3 (older version of this dataset with less inputs). The smallest datasets are provided to test more computationally demanding machine learning algorithms (e.g., SVM).

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

What type of data have you got for analysis?

We have chosen Bank Data for analysis which includes 41188 observations with 22 features, and the dataset is in CSV format.

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 a housing loan? (categorical: 'no','yes','unknown')

7 - loan: has a 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

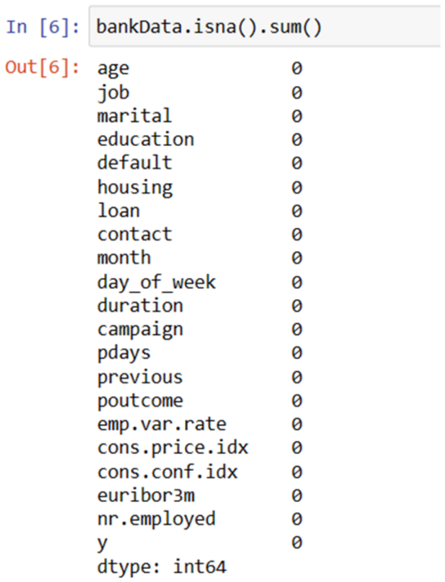
# other attributes:

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

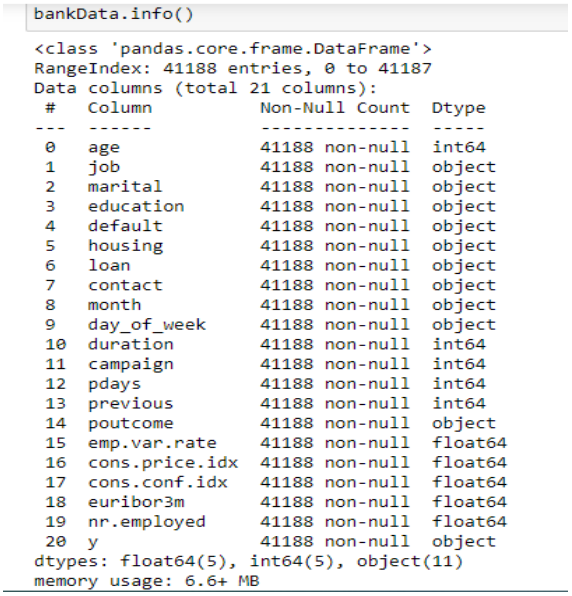
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success') # social and economic context attributes
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')

What are the problems in the data (number of NA values, outliers , skewed etc)?

1. Missing/NA Values:
    - We have investigated the dataset and checked if there are any missing values, utilizing the function called isna(). The dataset does not contain any missing values, which means there is no need for imputation or handling missing data.

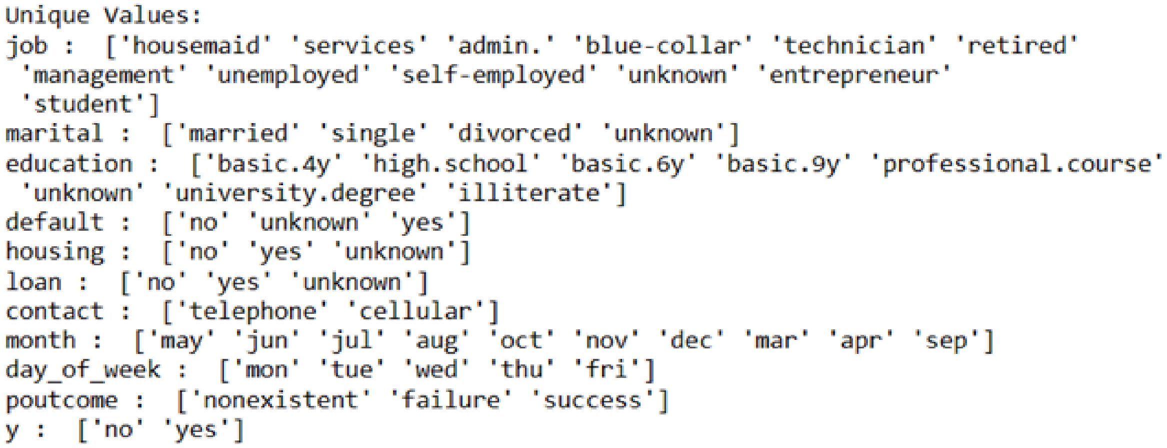


```
In [6]: bankData.isna().sum()

Out[6]: 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
```



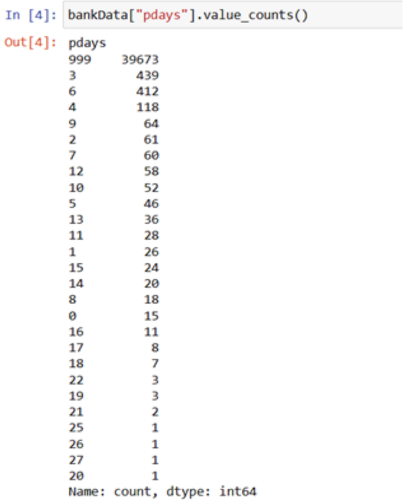
```
bankData.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             41188 non-null  int64  
 1   job             41188 non-null  object 
 2   marital         41188 non-null  object 
 3   education       41188 non-null  object 
 4   default         41188 non-null  object 
 5   housing         41188 non-null  object 
 6   loan            41188 non-null  object 
 7   contact         41188 non-null  object 
 8   month           41188 non-null  object 
 9   day_of_week     41188 non-null  object 
 10  duration        41188 non-null  int64  
 11  campaign        41188 non-null  int64  
 12  pdays           41188 non-null  int64  
 13  previous        41188 non-null  int64  
 14  poutcome        41188 non-null  object 
 15  emp.var.rate    41188 non-null  float64
 16  cons.price.idx  41188 non-null  float64
 17  cons.conf.idx   41188 non-null  float64
 18  euribor3m       41188 non-null  float64
 19  nr.employed     41188 non-null  float64
 20  y               41188 non-null  object 
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

2. Unknown Values in Categorical Columns:
    - Some categorical columns contain “unknown” values. These unknown values are a minority in the dataset and are not expected to have a significant impact on the models built. Therefore, they can be safely ignored without affecting the analysis.

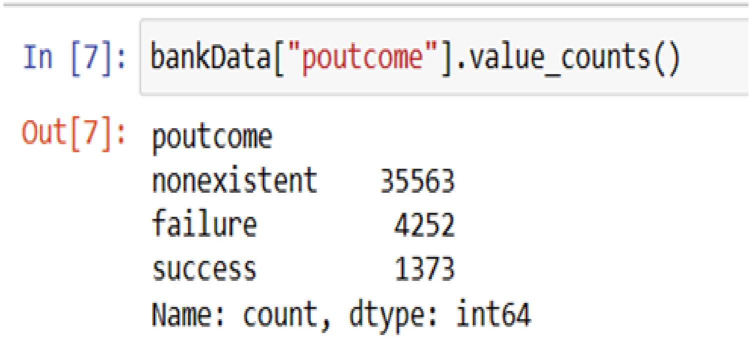


```
Unique Values:
job :  ['housemaid' 'services' 'admin.' 'blue-collar' 'technician' 'retired'
 'management' 'unemployed' 'self-employed' 'unknown' 'entrepreneur'
 'student']
marital :  ['married' 'single' 'divorced' 'unknown']
education :  ['basic.4y' 'high.school' 'basic.6y' 'basic.9y' 'professional.course'
 'unknown' 'university.degree' 'illiterate']
default :  ['no' 'unknown' 'yes']
housing :  ['no' 'yes' 'unknown']
loan :  ['no' 'yes' 'unknown']
contact :  ['telephone' 'cellular']
month :  ['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'mar' 'apr' 'sep']
day_of_week :  ['mon' 'tue' 'wed' 'thu' 'fri']
poutcome :  ['nonexistent' 'failure' 'success']
y :  ['no' 'yes']
```

3. Removal of "pdays" Column:
    - The "pdays" column has a majority of values set as 999. It appears that this value denotes a nonexistent or unknown outcome. Since there is already another column, "poutcome," which has a value "nonexistent" representing the same information, it is reasonable to remove the "pdays" column to avoid redundancy.

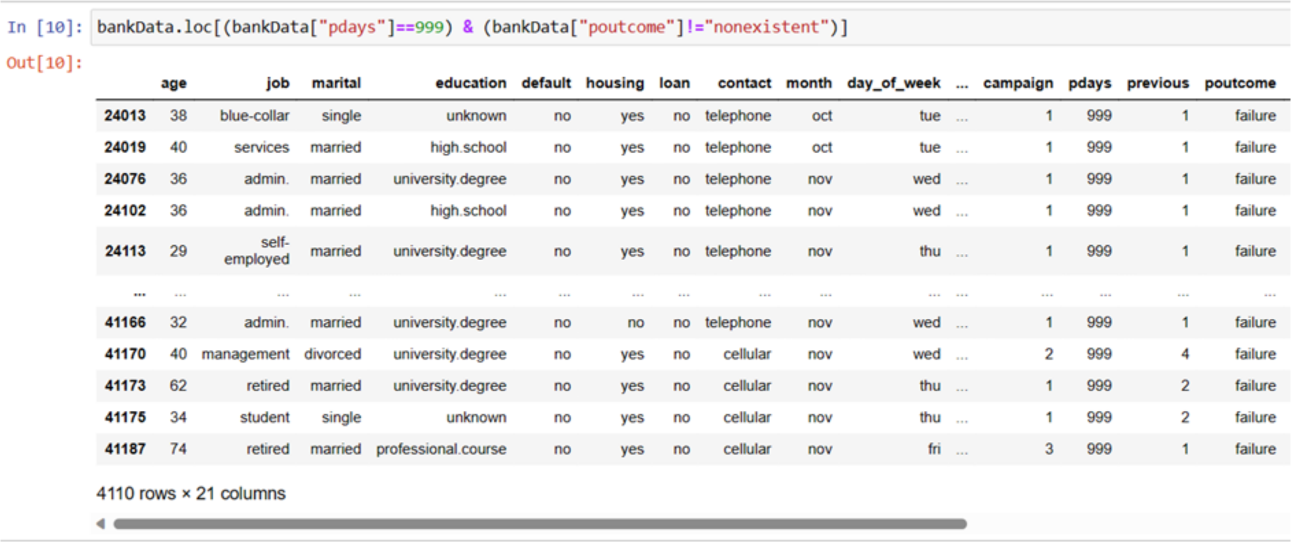


```
In [4]: bankData["pdays"].value_counts()

Out[4]: pdays
        999    39673
        3        439
        6        412
        4        118
        9         64
        2         61
        7         60
        12        58
        10        52
        5         46
        13        36
        11        28
        1         26
        15        24
        14        20
        8         18
        0         15
        16        11
        17         8
        18         7
        22         3
        19         3
        21         2
        25         1
        26         1
        27         1
        20         1
        Name: count, dtype: int64
```



```
In [7]: bankData["poutcome"].value_counts()

Out[7]: poutcome
        nonexistent    35563
        failure         4252
        success         1373
        Name: count, dtype: int64
```

It can be noted that there are 39673 rows where the value of "pdays" is 999. However, there are only 35563 rows where the value of "poutcome" is non-existent. This will have to be handled.



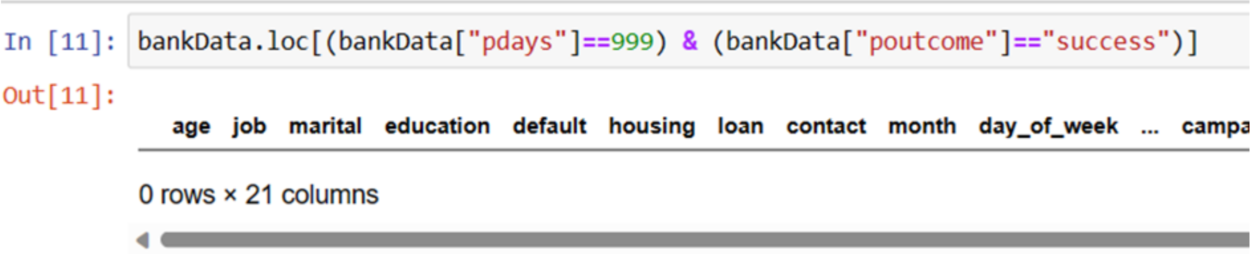
In [10]:
```
bankData.loc[(bankData["pdays"]==999) & (bankData["poutcome"]!="nonexistent")]
```

Out[10]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome
24013	38	blue-collar	single	unknown	no	yes	no	telephone	oct	tue	...	1	999	1	failure
24019	40	services	married	high.school	no	yes	no	telephone	oct	tue	...	1	999	1	failure
24076	36	admin.	married	university.degree	no	yes	no	telephone	nov	wed	...	1	999	1	failure
24102	36	admin.	married	high.school	no	yes	no	telephone	nov	wed	...	1	999	1	failure
24113	29	self-employed	married	university.degree	no	yes	no	telephone	nov	thu	...	1	999	1	failure
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
41166	32	admin.	married	university.degree	no	no	no	telephone	nov	wed	...	1	999	1	failure
41170	40	management	divorced	university.degree	no	yes	no	cellular	nov	wed	...	2	999	4	failure
41173	62	retired	married	university.degree	no	yes	no	cellular	nov	thu	...	1	999	2	failure
41175	34	student	single	unknown	no	yes	no	cellular	nov	thu	...	1	999	2	failure
41187	74	retired	married	professional.course	no	yes	no	cellular	nov	fri	...	3	999	1	failure

4110 rows × 21 columns

Upon further analysis it was found that for all such rows, the value of "poutcome" are failure.



In [11]:
```
bankData.loc[(bankData["pdays"]==999) & (bankData["poutcome"]=="success")]
```

Out[11]:

age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campa

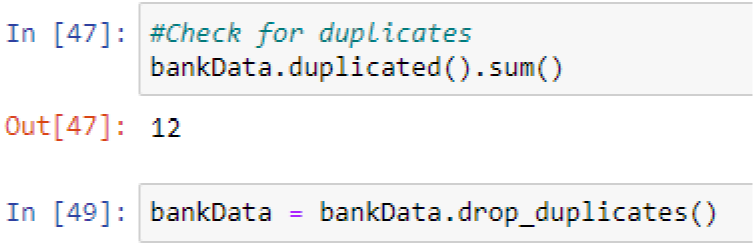
0 rows × 21 columns

This has to be further investigated to check if they were all true failures or have been mistakenly labeled failure instead of non-existent.

For the purposes of this project we have considered the latter.

4. Outliers:
    - The dataset does not have any significant outliers. Since there aren't any significant outliers in the dataset, there is no need for outlier treatment.

5. There were few duplicates in the dataset which was causing redundancy of the data.



```
In [47]: #Check for duplicates
         bankData.duplicated().sum()

Out[47]: 12

In [49]: bankData = bankData.drop_duplicates()
```

What approaches are you trying to apply on your data set to overcome problems like NA value, outlier etc and why?

Scaling of Numerical Values:

- Before building some machine learning models, the numerical values in the dataset have to be scaled. We plan to use a standard scaler to achieve this.

One-Hot Encoding of Categorical Values:

- To utilize the categorical columns in machine learning models, they need to be encoded numerically. The get_dummies function of Pandas will be used for the process

The duplicates were dropped to prevent redundancy:

- Removed duplicates with the drop_duplicates() function. This eliminated the duplicate rows in the entire dataset.

Github Repo link:

https://github.com/poojaravi05/Bank-Marketing-Campaign-