BANK MARKETING CAMPAIGN

Group Name

Sparagua

Name, Email:

- Daniela Alvarez

  daniela.alvarez04@gmail.com
- Akhil Nair

  akhil.nair1908@gmail.com

Country

- Peru
- India

College/Company

- Daniela: Datacamp, Kaggle Learn, Universidad de Piura (Peru)
- Akhil: SIESGST Nerul

Specialization:

Data Science

Problem description

Portuguese bank is having a decrease in its revenue. The bank wants to be able to predict which clients are most likely to subscribe a term deposit so they can focus

marketing efforts and resources on them and avoid wasting money and time on clients that will probably not subscribe.

Github Repo link

https://github.com/danielaaz04/Bank-Marketing-Campaign

Data cleansing and transformation done on the data.

Data Cleaning is a process of detecting and rectifying or deleting untrustworthy, inaccurate or outdated information from a dataset.

Steps followed for Bank Marketing Campaign cleaning:

1. Verify if there are any null values:

```
train.isnull().sum()
```

Results:

There were no features 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
```

2. Verify if there are any duplicated values:

```
train.duplicated().sum()
```

Results: There were 12 duplicated rows so we proceeded to drop them.

```
train.drop_duplicates(subset=None, inplace=True)
```

3. Look for strange values in categorical features:

```
print("Job:", train.job.value_counts(),sep = '\n')
```

We applied the previous command to all categorical features and found that features 'Job', 'Marital', 'Education', 'Default', 'Housing' and 'Loan' have an 'unknown' value. In the cleaning process we decided to use the mode or most frequent value to replace those unknow values.

```
def replace_with_frequent(df,col):
    frequent = df[col].value_counts().idxmax()
    print("The most frequent value is:", frequent)
    df[col].replace('unknown', frequent , inplace = True)
    print("Replacing unknown values with the most frequent value:", freque
nt)

#Replacing unknown values in categorical features.
replace_with_frequent(train, "job")
replace_with_frequent(train, "marital")
replace_with_frequent(train, "education")
replace_with_frequent(train, "default")
replace_with_frequent(train, "housing")
replace_with_frequent(train, "loan")
```

Results:

Unknown values in 'job' were replaced with 'admin'.

Unknown values in 'marital' were replaced with 'married'.

Unknown values in 'education' were replaced with 'university.degree'.

Unknown values in 'default' were replaced with 'no'.

Unknown values in 'housing' were replaced with 'yes'.

Unknown values in 'loan' were replaced with 'no'.

4. Replace unknown values in numerical features with the following function:

```
def replace_with_avg(df, col):
    average = df[col].mean(axis=0)
    print("The average is:" , average)
    df[col].replace('unknown', average , inplace = True)
    print("Replacing unknown values with average:", average)

#Replacing unknown values in numerical features.
replace_with_avg(train, "age")
replace_with_avg(train, "duration")
replace_with_avg(train, "campaign")
```

Results:

Unknown values in ‘age’ were replaced with ‘40’.

Unknown values in ‘duration’ were replaced with ‘258’.

Unknown values in ‘campaign’ were replaced with ‘2.56’.

5. After looking for outliers with the help of the describe() function, we noticed that there are three(3) features showing outliers: ‘age’, ‘duration’ and ‘campaign’.

We decided to use the IQR to establish upper and lower bounds so we can eliminate outliers.

The formula used was:

```
Stat = train[col].describe()

IQR = stat['75%']-stat['25%']
upper = stat['75%'] + 1.5*IQR
lower = stat['25%'] - 1.5*IQR
```

Results:

We did not have any lower outliers so there is no need to filter them, but there were upper outliers which we filtered to have a cleaner dataset with better statistics.

```
outliers_age = train[train['age'] >69.5]
outliers_duration =  train[train['duration'] >644.5]
outliers_campaign =  train[train['campaign'] >6]
```

Our new train dataset would now be:

```
new_train = train[(train["age"] < 69.5) & (train["duration"] < 644.5) & (t
rain["campaign"] < 6)]
```

Conclusion:

For a better modeling we have eliminated duplicated rows, we have replaced numerical unknown values with feature average, we have replaced categorical unknown values with Mode, and we have eliminated outliers using the IQR. After all these cleaning the shape of our new data is 34623 rows.

In the next report we will use visualizations to make a better exploratory data analysis. We will decide if eliminating 'Duration' feature as suggested, and we will encode our categorical values to get them ready for modeling.