Week 9: Deliverables – Data Cleaning Report

Group Name: AI Boys

Date: 12/02/2022

Group Member ID	Name	Email	Country	College	Specialization
1	Islom Pulatov	islompulatov115@gmail.com	Poland	Epicode Global	Data Science
2	Ammar Sidhu	ammarsidhu@outlook.com	Canada	University of Toronto	Data Science

Project: Bank Marketing (Campaign)

Github Repo: https://github.com/Islompulatov/Bank_marketing

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).

Business 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, 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).

Data Cleaning:

- Found **no missing values** in the dataset
- Found **no duplicate rows** in the dataset
- Would have handled missing values with two different approaches if present:
  (1) **Dropped** Missing Values (Ammar)
  (2) **Impute** Missing Values with **Median** (Islom)
- Would have **dropped duplicate rows** in the dataset if present
- Handled outliers for the **7 Numerical Features** with two different approaches:
  (1) **Do not drop** outliers (Islom)
  (2) **Drop** outliers based on feature and context of the outliers (Ammar)

Data Cleaning Results:

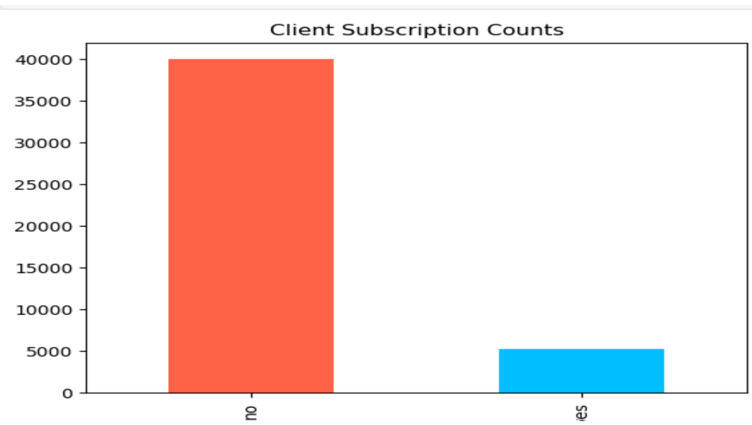
(1) Unique Values by Feature

	# of Unique Values:
age	77
job	12
marital	3
education	4
default	2
balance	7168
housing	2
loan	2
contact	3
day	31
month	12
duration	1573
campaign	48
pdays	559
previous	41
poutcome	4
y	2

(2) Summary Statistics for Numerical Features

	count	mean	std	min	25%	50%	75%	max
age	45211.0	40.936210	10.618762	18.0	33.0	39.0	48.0	95.0
balance	45211.0	1362.272058	3044.765829	-8019.0	72.0	448.0	1428.0	102127.0
day	45211.0	15.806419	8.322476	1.0	8.0	16.0	21.0	31.0
duration	45211.0	258.163080	257.527812	0.0	103.0	180.0	319.0	4918.0
campaign	45211.0	2.763841	3.098021	1.0	1.0	2.0	3.0	63.0
pdays	45211.0	40.197828	100.128746	-1.0	-1.0	-1.0	-1.0	871.0
previous	45211.0	0.580323	2.303441	0.0	0.0	0.0	0.0	275.0

(3) Visualizing Target Feature



(4) Missing Values by Feature

```
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 (0) missig values** in this dataset!

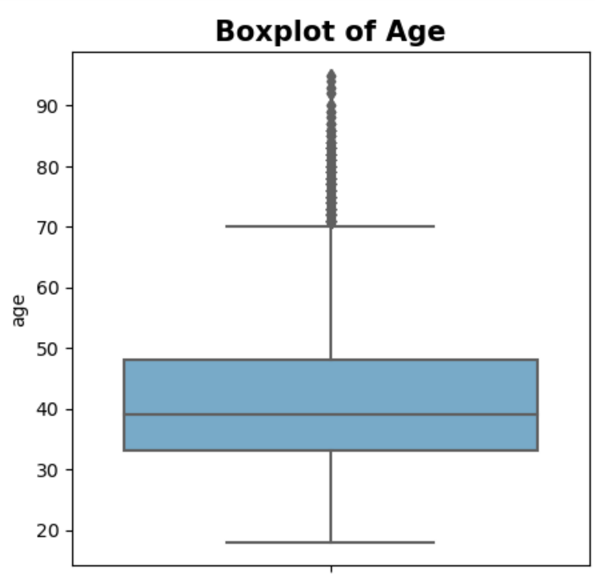
(5) Duplicate Rows in Dataset

```python
# Number of duplicates
duplicates_number = df.duplicated().sum()
print("Number of duplicated rows is: ", duplicates_number)
```

```
Number of duplicated rows is:  0
```

There are **no (0) duplicate rows** in this dataset!

(6) Outliers in *age* Feature

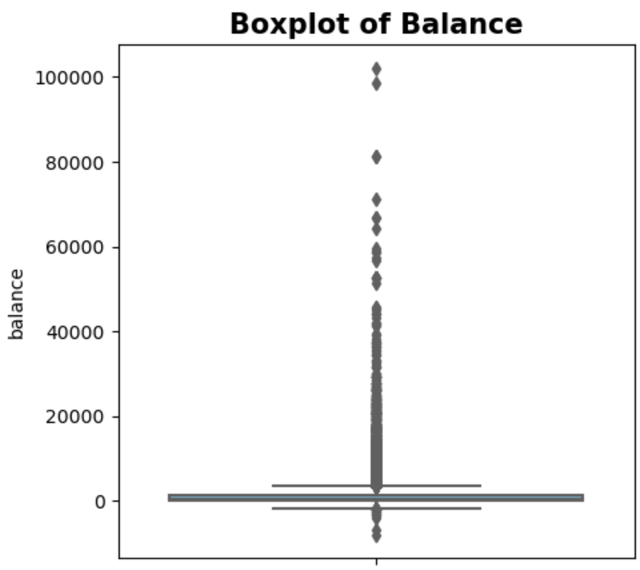



```python
# Using IQR to inspect outliers
age_stats = df['age'].describe()
IQR = age_stats['75%'] - age_stats['25%']
upper_bound = age_stats['75%'] + 1.5 * IQR
lower_bound = age_stats['25%'] - 1.5 * IQR
print("The upper and lower bounds for the age feature are: ", (upper_bound, lower_bound))
```

```
The upper and lower bounds for the age feature are:  (70.5, 10.5)
```

The `age` feature has outliers in the upper bound because of the eldery population who are a legitmate representation of the population of the customer. Removing them does not make sense.

(7) Outliers in *balance* Feature

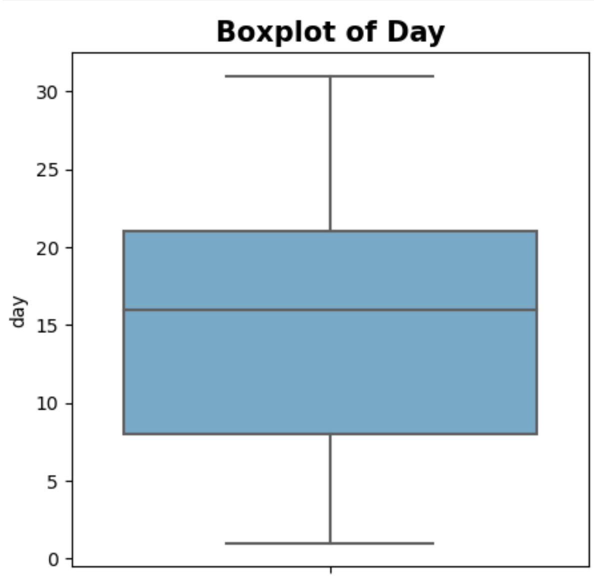



```
# Using IQR to inspect outliers
balance_stats = df['balance'].describe()
IQR = balance_stats['75%'] - balance_stats['25%']
upper_bound = balance_stats['75%'] + 1.5 * IQR
lower_bound = balance_stats['25%'] - 1.5 * IQR
print("The upper and lower bounds for the balance feature are: ", (upper_bound, lower_bound))
```

```
The upper and lower bounds for the balance feature are:  (3462.0, -1962.0)
```

We will remove the outliers in the lower and upper bound of the `balance` feature.

(8) Outliers in *day* Feature

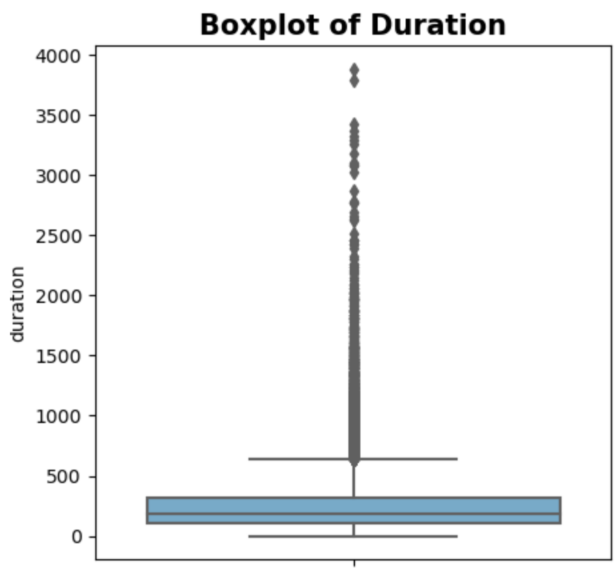


```python
# Using IQR to inspect outliers
day_stats = df['day'].describe()
IQR = day_stats['75%'] - day_stats['25%']
upper_bound = day_stats['75%'] + 1.5 * IQR
lower_bound = day_stats['25%'] - 1.5 * IQR
print("The upper and lower bounds for the day feature are: ", (upper_bound, lower_bound))
```

```
The upper and lower bounds for the day feature are:  (40.5, -11.5)
```

There are no outliers present in the `day` feature.

(9) Outliers in *duration* feature




```python
# Using IQR to inspect outliers
duration_stats = df['duration'].describe()
IQR = duration_stats['75%'] - duration_stats['25%']
upper_bound = duration_stats['75%'] + 1.5 * IQR
lower_bound = duration_stats['25%'] - 1.5 * IQR
print("The upper and lower bounds for the duration feature are: ", (upper_bound, lower_bound))
```

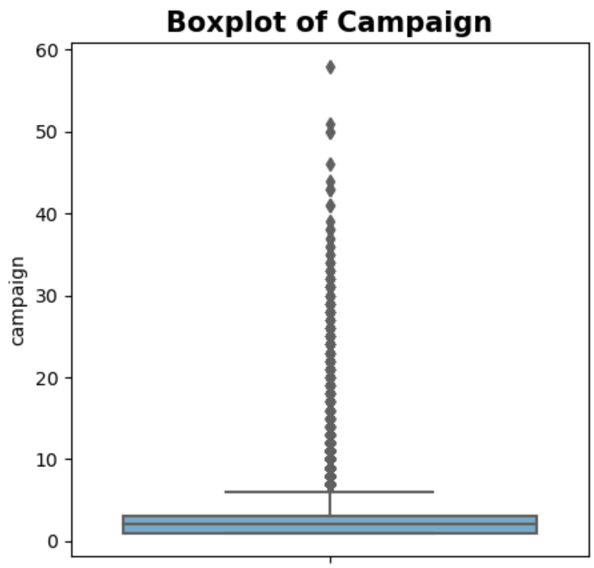
```
The upper and lower bounds for the duration feature are:  (635.5, -216.5)
```

We will remove the outliers in the lower and upper bound of the `duration` feature.

```python
# Drop Outliers
df.drop(df[df['duration'] > upper_bound].index, inplace = True)
df.drop(df[df['duration'] < lower_bound].index, inplace = True)
print(df.shape)
```

```
(37572, 17)
```

(10) Outliers in *campaign* feature

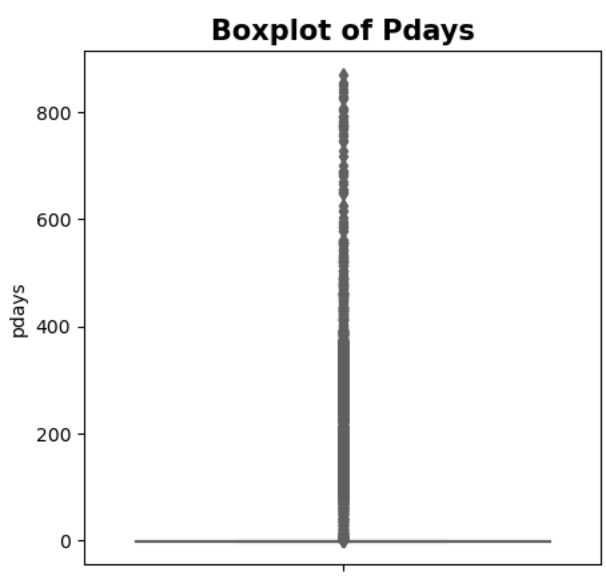



```
# Using IQR to inspect outliers
campaign_stats = df['campaign'].describe()
IQR = campaign_stats['75%'] - campaign_stats['25%']
upper_bound = campaign_stats['75%'] + 1.5 * IQR
lower_bound = campaign_stats['25%'] - 1.5 * IQR
print("The upper and lower bounds for the campaign feature are: ", (upper_bound, lower_bound))
```

```
The upper and lower bounds for the campaign feature are:  (6.0, -2.0)
```

We will remove the outliers in the upper bound of the `campaign` feature.

(11) Outliers in *pdays* feature

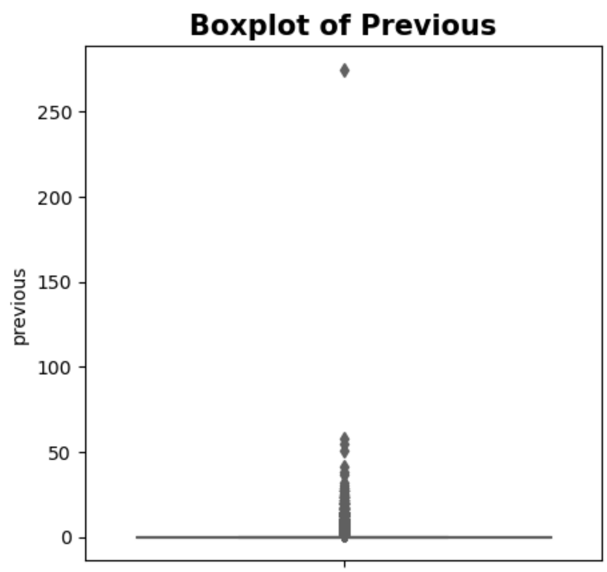


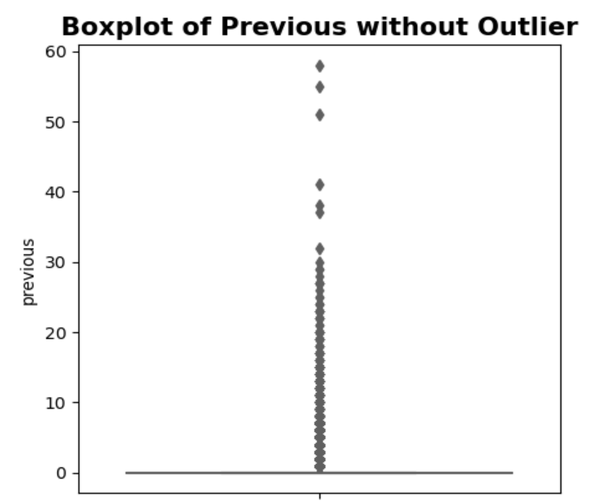
```
# Using IQR to inspect outliers
pdays_stats = df['pdays'].describe()
IQR = pdays_stats['75%'] - pdays_stats['25%']
upper_bound = pdays_stats['75%'] + 1.5 * IQR
lower_bound = pdays_stats['25%'] - 1.5 * IQR
print("The upper and lower bounds for the pdays feature are: ", (upper_bound, lower_bound))
```

```
The upper and lower bounds for the pdays feature are:  (-1.0, -1.0)
```

The outliers in the `pday` feature will not be removed. There are a lot of clients that haven't been contacted for a while, but it does not make sense to remove them becuase it is important to understand their inactvity. They definitely play a significant role in the context of the data as they represent a chunk of the client population.

(12) Outliers in *previous* feature






```
# Using IQR to inspect outliers
previous_stats = df['previous'].describe()
IQR = previous_stats['75%'] - previous_stats['25%']
upper_bound = previous_stats['75%'] + 1.5 * IQR
lower_bound = previous_stats['25%'] - 1.5 * IQR
print("The upper and lower bounds for the previous feature are: ", (upper_bound, lower_bound))
```

```
The upper and lower bounds for the previous feature are:  (0.0, 0.0)
```

Based on our boxplot, there is one apparent outlier that crosses 250 in the `previous` feature. This appears to be a very loyal customer that has conducted a lot of campaigns prior with the company. However, this client is clearly an outlier. The other customers that lie above the upper bound will not be dropped because they make sense as there are going to be a lot of returning customers who have been signing contracts with the company in the past.

Next Steps:

- Complete Data Analysis, Data Visualizations, and Modeling with Outliers (Islom)
- Complete Data Analysis, Data Visualizations, and Modeling without Outliers (Ammar)