

DELIVERABLE WEEK – 9(1)

Specialization : Data Science

Name : Mohammed Maqsood
Email : mohammedmaqsood48@gmail.com
Country : Germany
College : Otto von Guericke University
Specialization : Chemical and Energy Engineering

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 helps 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).

Here we are using different approaches to clean and transform the data in order to solve the above-mentioned problem.

GitHub Repo link

<https://github.com/Maqsood8/Group-Project-Bank-Marketing.git>

Data Cleansing and Transformation

Results and Approaches:

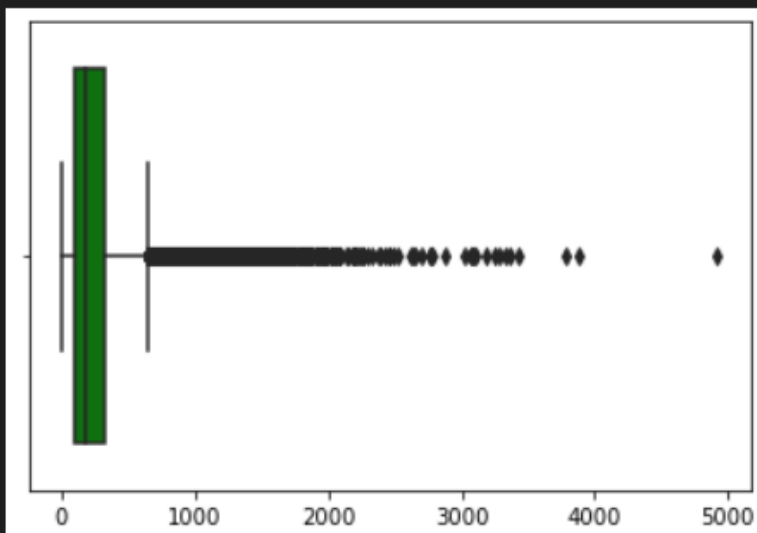
- Remove the quotes in the values of the data
- No missing data
- No duplicated values
- Provide the appropriate column name to the data
- Provide the correct data type to each column
- All the unknown data has been deleted because they are considered as missing value
- Calculate the skewed value of each numerical value
- Check outliers in the data

Outliers

The image below shows all the data columns that has outliers, but we decided to keep them because we can see certain number of Outliers in 'age', 'duration', and 'campaign' etc.

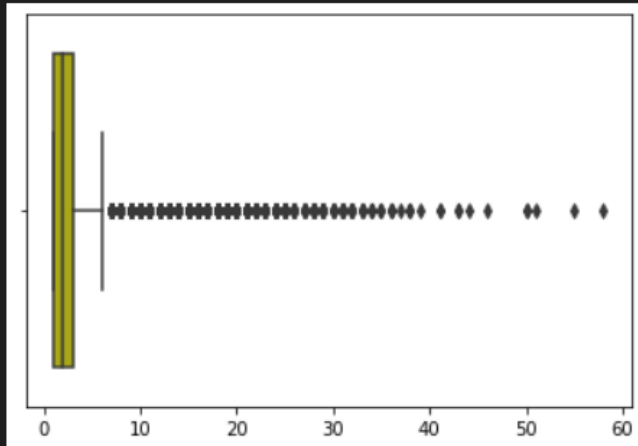
```
sns.boxplot(bank.duration.values,color = 'g')
```

```
C:\Users\prani\anaconda3\lib\site-packages\seaborn\_decorators.py:100: UserWarning:
The `data` argument will be `data`, and passing other arguments will be deprecated.
warnings.warn(
<AxesSubplot:>
```



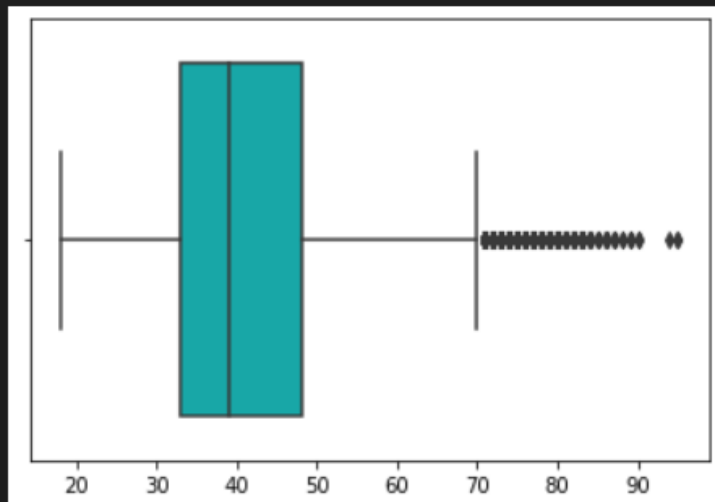
```
sns.boxplot(bank.campaign.values, color = 'y')
```

C:\Users\prani\anaconda3\lib\site-packages\seaborn_decorat
argument will be `data`, and passing other arguments without
warnings.warn(
<AxesSubplot: >



```
sns.boxplot(bank.age.values, color = 'c')
```

C:\Users\prani\anaconda3\lib\site-packages\seaborn_d
argument will be `data`, and passing other arguments
warnings.warn(
<AxesSubplot: >



But it's important to note that since this is a sensitive Bank Dataset the above columns should be treated as 'Extreme values' which provides important insights and not 'Outliers'

Skewed

```
cols=["age","duration","campaign","pdays","previous","balance"]
for i in cols:
    print(f"Skewness {i} : " + str(bank[i].skew()))
```

```
... Skewness age : 0.6978356364509636
Skewness duration : 3.1701799697784785
Skewness campaign : 4.7924941810208885
Skewness pdays : 2.608337543002269
Skewness previous : 42.08877792244101
Skewness balance : 8.400120937754398
```

The most skewed values are positive that means mean and median of data is greater than mode and also previous has highest skewness.