Deliverable – Week 9

Group Name: Tony's group

Team:

Name: Tony Nguyen

Email: tonynguyen9707@gmail.com

Country: United States

College: University of Florida

Specialization: Data Analyst

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).

GitHub Link

https://github.com/ttn20/DataGlacierInternship/tree/Bank-Marketing-Final-Project

Data Cleaning and Transformation

- Transformed data to include appropriate column names.
- Transformed data to include appropriate data types to each column.
- Fixed formatting of data (issue with separation between variables).
- Found no missing data.
- Found issue with unknown data. Solved: Deleted unknown data.
- Multiple methods to check for outliers and skewness.

Issues with data & solutions

```
# Checking for Outliers & Skewness
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	43193.0	40.764082	10.512640	18.0	33.0	39.0	48.0	95.0
balance	43193.0	1354.027342	3042.103625	-8019.0	71.0	442.0	1412.0	102127.0
day	43193.0	15.809414	8.305970	1.0	8.0	16.0	21.0	31.0
duration	43193.0	258.323409	258.162006	0.0	103.0	180.0	318.0	4918.0
campaign	43193.0	2.758178	3.063987	1.0	1.0	2.0	3.0	58.0
pdays	43193.0	40.404070	100.420624	-1.0	-1.0	-1.0	-1.0	871.0
previous	43193.0	0.584863	2.332672	0.0	0.0	0.0	0.0	275.0

Age: Looks normal/reasonable.

Balance: Completely skewed with outliers.

Day: Looks normal/reasonable.

Duration: Completely skewed with outliers.

Campaign: Completely skewed with outliers.

Pdays: Potential issue.

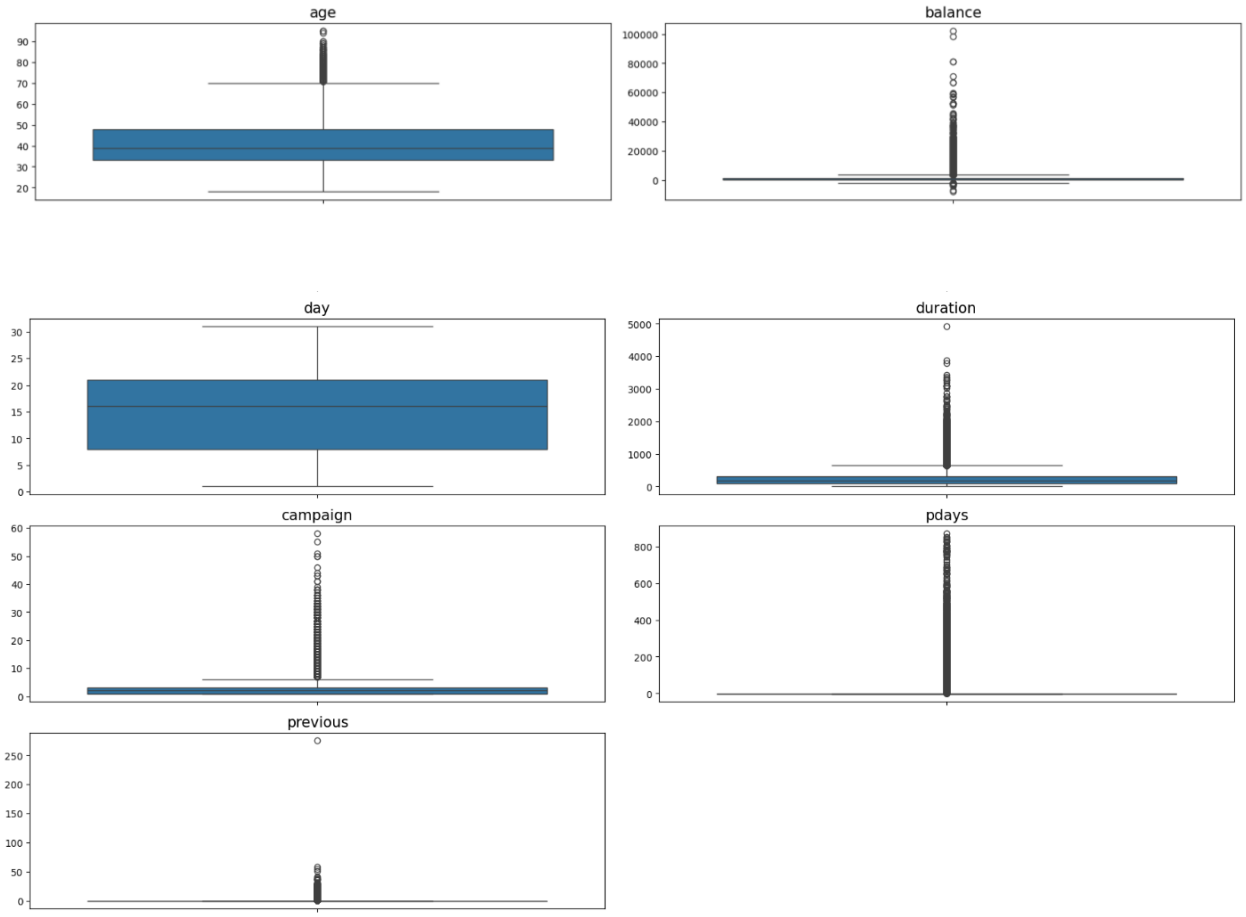
Previous: Completely skewed with outliers.

It is worth noting that despite being outliers, these values are crucial as the data is sensitive which means the outliers could provide useful information.

```python
# Alternative method for checking for outliers and skewness
fig, ax = plt.subplots(5, 2, figsize=(18, 15))
count = 0
cols = data.select_dtypes(include=np.number).columns
total_cols = len(cols)

for i in range(5):
    for j in range(2):
        if count < total_cols:
            s = cols[count]
            sb.boxplot(data[s].values, ax=ax[i][j], orient='vertical')
            ax[i][j].set_title(s, fontsize=15)
            count += 1
        else:
            fig.delaxes(ax[i][j])

fig.set_size_inches(18, 15)
plt.tight_layout()
plt.show()
```



Similar results as previous method.

```
# Alternative method for checking for skewness


for i in int_cols:
    print(f"Skewness {i} : " + str(data[i].skew()))
```

```
Skewness age : 0.6978356364509636
Skewness balance : 8.400120937754398
Skewness day : 0.08979984840490052
Skewness duration : 3.1701799697784785
Skewness campaign : 4.7924941810208885
Skewness pdays : 2.608337543002269
Skewness previous : 42.08877792244101
```

Values that are not near 0 are considered skewed as 0 represents symmetrical distribution.