Cleaning Bank Deposit Data

Presentation of Data Set

- Format

  11222 rows × 17 columns

- Type

  numeric: age, balance, day, duration, campaign, pdays, previous

  categorical: job, marital, education, contact, month

  binary: default, housing, loan, poutcome,Bank deposit(target)

Reasons to hate the data provider

- No Primary Key

Impossibility to spot duplicates

- Age instead of birthday
- Inconsistency with data descritption

People mentionned as never been called still have a non null duration of last call

- Number of unknown values

Poutcome (previous outcome) could be useful, but thus had to be dropped

Missing data

1. Spot and count missing values

   df0.isnull().sum()

   output:113 missing values over 11K+ rows. Not too bad

2. Treatment

   delete row if too many missing values in the row

   replace nan by mode of the series (for ex: 6 missing values in Previous, and 9211 0=> replace missing by 0)

Outliers

1. Describe the data set

data1.describe().transpose()

	count	mean	std	min	25%	50%	75%	max
CustomerID	11222.0	5610.500000	3239.656695	0.0	2805.25	5610.5	8415.75	11221.0
age	11222.0	56.411068	6.141462	50.0	52.00	55.0	58.00	95.0
balance	11216.0	7966.974412	642145.646918	-4057.0	108.00	627.5	2031.75	68000000.0
day	11216.0	15.786912	8.336913	1.0	8.00	16.0	21.00	31.0
duration	11214.0	2040.672106	188861.984549	0.0	102.00	176.0	316.00	20000000.0
campaign	11214.0	2.737739	2.854410	1.0	1.00	2.0	3.00	43.0
pdays	11214.0	35.118245	90.776604	-1.0	-1.00	-1.0	-1.00	792.0
previous	11216.0	0.558934	1.741345	0.0	0.00	0.0	0.00	37.0

A couple rows to get rid of.

Visualization

pyplot / seaborn

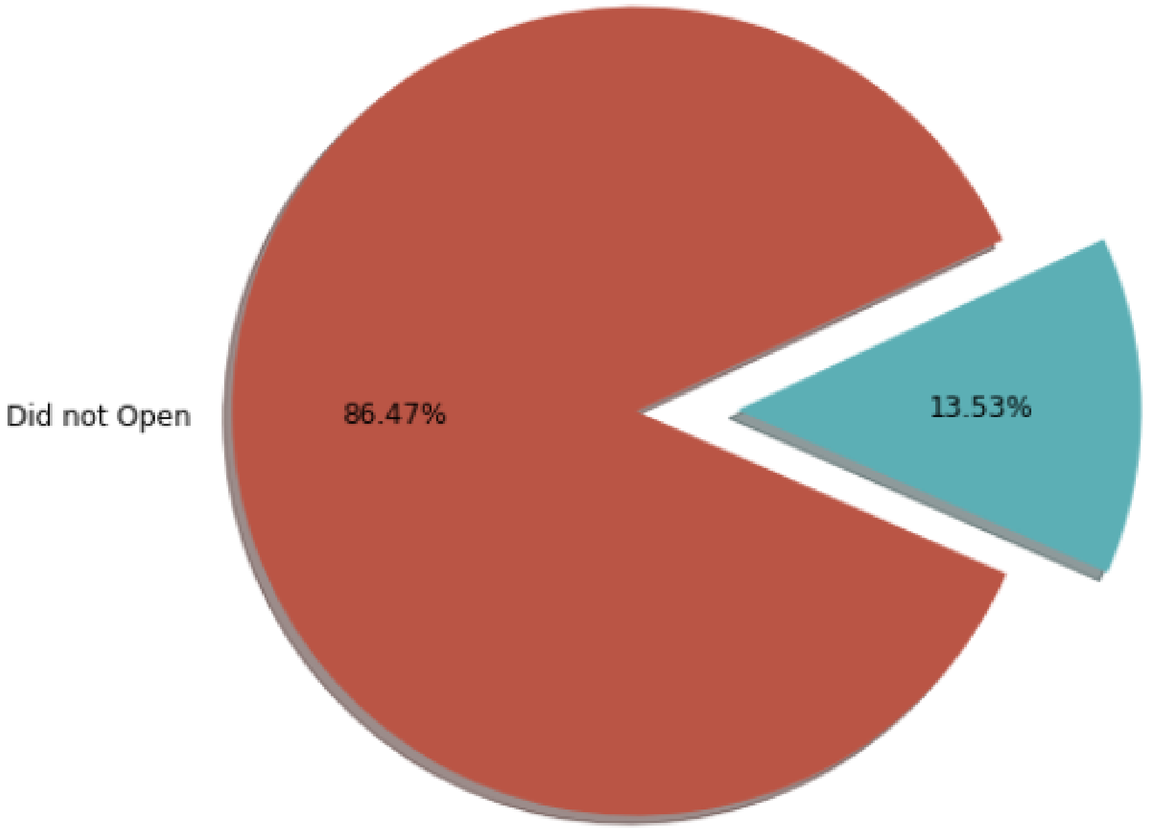
Used those libraries
due to their ease of use
over efficiency ratio

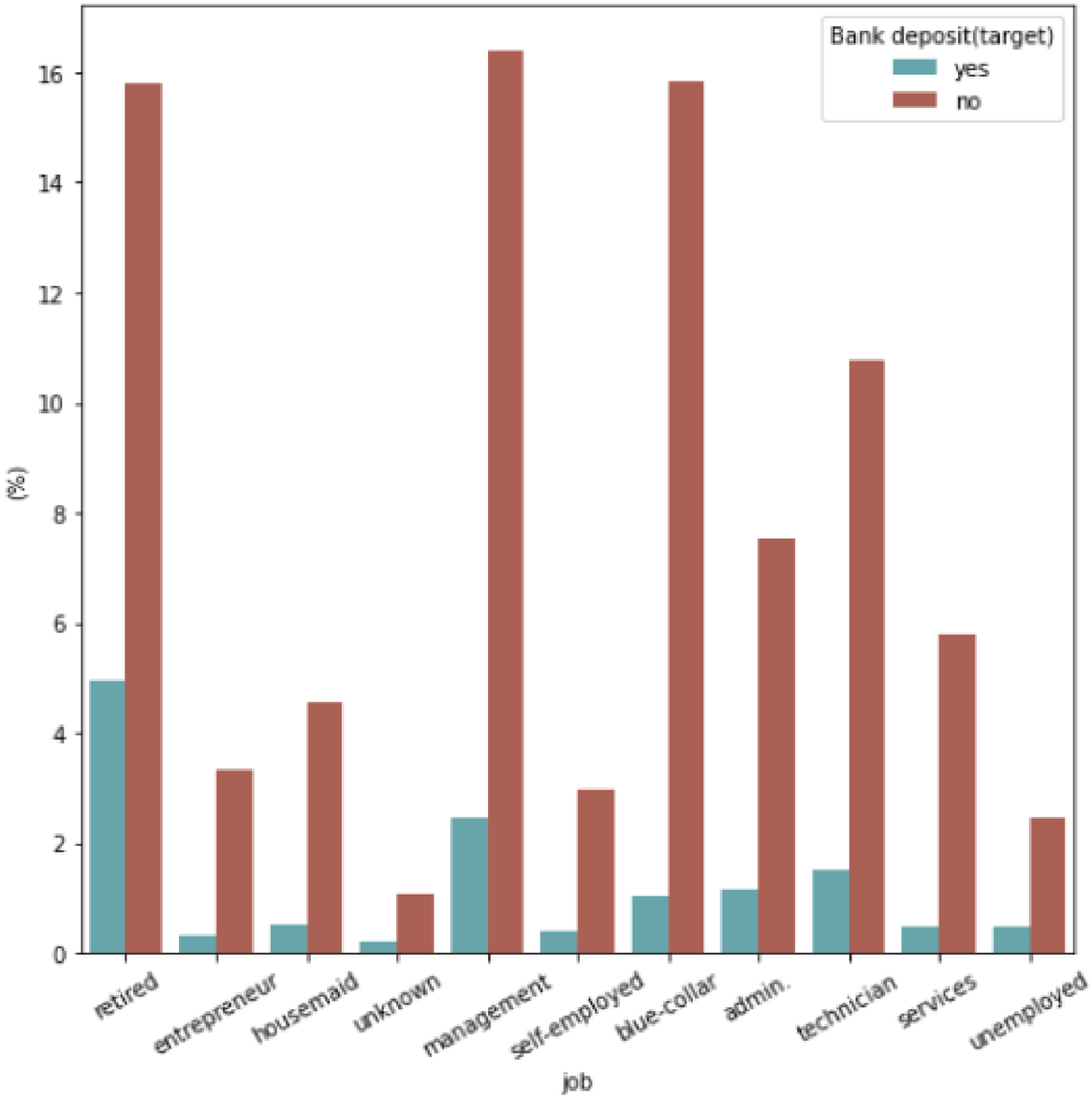
Tons of parameters
to make the data
look just nice

```
f, ax = plt.subplots(1,2, figsize=(18,8))

colors = ["#ba5545", "#5cafb5"]
labels ="Did not Open", "Opened"

plt.suptitle('Success/Failure analysis', fontsize=20)

data1["Bank deposit(target)"].value_counts().plot.pie(
explode=[0,0.25], autopct='%1.2f%%', ax=ax[0],shadow=True,
colors=colors, labels=labels, fontsize=12, startangle=25)

ax[0].set_ylabel('', fontsize=14, )

palette = ["#5cafb5", "#ba5545"]

sns.barplot(x="job", y="balance", hue="Bank deposit(target)",
data=data1, palette=palette,
estimator=lambda x: len(x) / len(data1) * 100)
ax[1].set(ylabel="(%)")
ax[1].set_xticklabels(data1["job"].unique(), rotation=30)
plt.show()
```

Visualization

Success/Failure analysis





Success rate is low, but still higher when it comes to retired people

Dive deeper into clients' profile

SQL Query:

```
CREATE TABLE average_data (SELECT job,
    ROUND(AVG(age), 2) AS avg_age,
    ROUND(AVG(duration), 2) AS avg_duration,
    ROUND(AVG(balance), 2) AS avg_balance FROM
    bank_deposit
WHERE
    target = 'yes'
GROUP BY job);
```

Quantitative data about the data set

job	avg_age	avg_duration	avg_balance
retired	67.45	457.43	2385.42
entrepreneur	56.17	553.86	2255.6
housemaid	59.61	566.08	1727.75
unknown	57.05	444.36	3258.59
management	56.71	481.41	2380.34
self-employed	58.93	451.48	4808.43
admin.	55.86	522.43	2332.94
technician	55.38	509.28	1771.48
unemployed	55.02	516.5	1243.3
blue-collar	54.79	645.15	1859.56
services	54.29	570.16	1147.31