In [86]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("bank.csv", delimiter = ";")
# df = pd.read_csv("Datasets//bank//bank.csv", delimiter = ";")
df.head()
```

Out[86]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	du
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	

In [87]:
```
df.isna().sum().sum()
```

Out[87]:
```
0
```

In [88]:
```
df.isna().sum()
```

Out[88]:
```
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
```

In [89]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        4521 non-null   int64 
 1   job        4521 non-null   object
 2   marital    4521 non-null   object
 3   education  4521 non-null   object
 4   default    4521 non-null   object
 5   balance    4521 non-null   int64 
 6   housing    4521 non-null   object
 7   loan       4521 non-null   object
 8   contact    4521 non-null   object
 9   day        4521 non-null   int64 
 10  month      4521 non-null   object
 11  duration   4521 non-null   int64 
 12  campaign   4521 non-null   int64 
 13  pdays      4521 non-null   int64 
 14  previous   4521 non-null   int64 
 15  poutcome   4521 non-null   object
 16  y          4521 non-null   object
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
```

In [90]:
```
df.describe()
```

Out[90]:

	age	balance	day	duration	campaign	pdays	previous
count	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000
mean	41.170095	1422.657819	15.915284	263.961292	2.793630	39.766645	0.542579
std	10.576211	3009.638142	8.247667	259.856633	3.109807	100.121124	1.693562
min	19.000000	-3313.000000	1.000000	4.000000	1.000000	-1.000000	0.000000
25%	33.000000	69.000000	9.000000	104.000000	1.000000	-1.000000	0.000000
50%	39.000000	444.000000	16.000000	185.000000	2.000000	-1.000000	0.000000
75%	49.000000	1480.000000	21.000000	329.000000	3.000000	-1.000000	0.000000
max	87.000000	71188.000000	31.000000	3025.000000	50.000000	871.000000	25.000000

In [14]:
```
!pip install pydantic-settings
!pip install ydata-profiling
```

```
Collecting pydantic-settings
  Obtaining dependency information for pydantic-settings from https://files.pythonhos
ted.org/packages/99/ee/24ec87e3a91426497c5a2b9880662d19cfd640342d477334ebc60fc2c276/p
ydantic_settings-2.2.1-py3-none-any.whl.metadata
  Downloading pydantic_settings-2.2.1-py3-none-any.whl.metadata (3.1 kB)
Requirement already satisfied: pydantic>=2.3.0 in c:\users\zhiyan\anaconda3\lib\site-
packages (from pydantic-settings) (2.6.2)
Requirement already satisfied: python-dotenv>=0.21.0 in c:\users\zhiyan\anaconda3\lib
\site-packages (from pydantic-settings) (0.21.0)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\zhiyan\anaconda3\li
b\site-packages (from pydantic>=2.3.0->pydantic-settings) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in c:\users\zhiyan\anaconda3\lib
\site-packages (from pydantic>=2.3.0->pydantic-settings) (2.16.3)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\zhiyan\anaconda3
\lib\site-packages (from pydantic>=2.3.0->pydantic-settings) (4.7.1)
Downloading pydantic_settings-2.2.1-py3-none-any.whl (13 kB)
Installing collected packages: pydantic-settings
Successfully installed pydantic-settings-2.2.1
Requirement already satisfied: ydata-profiling in c:\users\zhiyan\anaconda3\lib\site-
packages (4.6.4)
Requirement already satisfied: scipy<1.12,>=1.4.1 in c:\users\zhiyan\anaconda3\lib\si
te-packages (from ydata-profiling) (1.11.1)
Requirement already satisfied: pandas!=1.4.0,<3,>1.1 in c:\users\zhiyan\anaconda3\lib
\site-packages (from ydata-profiling) (2.0.3)
Requirement already satisfied: matplotlib<3.9,>=3.2 in c:\users\zhiyan\anaconda3\lib
\site-packages (from ydata-profiling) (3.7.2)
Requirement already satisfied: pydantic>=2 in c:\users\zhiyan\anaconda3\lib\site-pack
ages (from ydata-profiling) (2.6.2)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in c:\users\zhiyan\anaconda3\lib\si
te-packages (from ydata-profiling) (6.0)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in c:\users\zhiyan\anaconda3\lib\s
ite-packages (from ydata-profiling) (3.1.2)
Requirement already satisfied: visions[type_image_path]==0.7.5 in c:\users\zhiyan\ana
conda3\lib\site-packages (from ydata-profiling) (0.7.5)
Requirement already satisfied: numpy<1.26,>=1.16.0 in c:\users\zhiyan\anaconda3\lib\s
ite-packages (from ydata-profiling) (1.24.3)
Requirement already satisfied: htmlmin==0.1.12 in c:\users\zhiyan\anaconda3\lib\site-
packages (from ydata-profiling) (0.1.12)
Requirement already satisfied: phik<0.13,>=0.11.1 in c:\users\zhiyan\anaconda3\lib\si
te-packages (from ydata-profiling) (0.12.4)
Requirement already satisfied: requests<3,>=2.24.0 in c:\users\zhiyan\anaconda3\lib\s
ite-packages (from ydata-profiling) (2.31.0)
Requirement already satisfied: tqdm<5,>=4.48.2 in c:\users\zhiyan\anaconda3\lib\site-
packages (from ydata-profiling) (4.65.0)
Requirement already satisfied: seaborn<0.13,>=0.10.1 in c:\users\zhiyan\anaconda3\lib
\site-packages (from ydata-profiling) (0.12.2)
Requirement already satisfied: multimethod<2,>=1.4 in c:\users\zhiyan\anaconda3\lib\s
ite-packages (from ydata-profiling) (1.11.1)
Requirement already satisfied: statsmodels<1,>=0.13.2 in c:\users\zhiyan\anaconda3\li
b\site-packages (from ydata-profiling) (0.14.0)
Requirement already satisfied: typeguard<5,>=4.1.2 in c:\users\zhiyan\anaconda3\lib\s
ite-packages (from ydata-profiling) (4.1.5)
Requirement already satisfied: imagehash==4.3.1 in c:\users\zhiyan\anaconda3\lib\site
-packages (from ydata-profiling) (4.3.1)
Requirement already satisfied: wordcloud>=1.9.1 in c:\users\zhiyan\anaconda3\lib\site
-packages (from ydata-profiling) (1.9.3)
Requirement already satisfied: dacite>=1.8 in c:\users\zhiyan\anaconda3\lib\site-pack
ages (from ydata-profiling) (1.8.1)
Requirement already satisfied: numba<0.59.0,>=0.56.0 in c:\users\zhiyan\anaconda3\lib
\site-packages (from ydata-profiling) (0.57.1)
```

```
Requirement already satisfied: PyWavelets in c:\users\zhiyan\anaconda3\lib\site-packa
ges (from imagehash==4.3.1->ydata-profiling) (1.4.1)
Requirement already satisfied: pillow in c:\users\zhiyan\anaconda3\lib\site-packages
(from imagehash==4.3.1->ydata-profiling) (9.4.0)
Requirement already satisfied: attrs>=19.3.0 in c:\users\zhiyan\anaconda3\lib\site-pa
ckages (from visions[type_image_path]==0.7.5->ydata-profiling) (22.1.0)
Requirement already satisfied: networkx>=2.4 in c:\users\zhiyan\anaconda3\lib\site-pa
ckages (from visions[type_image_path]==0.7.5->ydata-profiling) (3.1)
Requirement already satisfied: tangled-up-in-unicode>=0.0.4 in c:\users\zhiyan\anacon
da3\lib\site-packages (from visions[type_image_path]==0.7.5->ydata-profiling) (0.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\zhiyan\anaconda3\lib\site-
packages (from jinja2<3.2,>=2.11.1->ydata-profiling) (2.1.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\zhiyan\anaconda3\lib\site
-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\zhiyan\anaconda3\lib\site-pac
kages (from matplotlib<3.9,>=3.2->ydata-profiling) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\zhiyan\anaconda3\lib\sit
e-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\zhiyan\anaconda3\lib\sit
e-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\zhiyan\anaconda3\lib\site-
packages (from matplotlib<3.9,>=3.2->ydata-profiling) (23.1)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\zhiyan\anaconda3\lib
\site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\zhiyan\anaconda3\lib
\site-packages (from matplotlib<3.9,>=3.2->ydata-profiling) (2.8.2)
Requirement already satisfied: llvmlite<0.41,>=0.40.0dev0 in c:\users\zhiyan\anaconda
3\lib\site-packages (from numba<0.59.0,>=0.56.0->ydata-profiling) (0.40.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\zhiyan\anaconda3\lib\site-pac
kages (from pandas!=1.4.0,<3,>1.1->ydata-profiling) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\zhiyan\anaconda3\lib\site-p
ackages (from pandas!=1.4.0,<3,>1.1->ydata-profiling) (2023.3)
Requirement already satisfied: joblib>=0.14.1 in c:\users\zhiyan\anaconda3\lib\site-p
ackages (from phik<0.13,>=0.11.1->ydata-profiling) (1.1.1)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\zhiyan\anaconda3\li
b\site-packages (from pydantic>=2->ydata-profiling) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in c:\users\zhiyan\anaconda3\lib
\site-packages (from pydantic>=2->ydata-profiling) (2.16.3)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\zhiyan\anaconda3
\lib\site-packages (from pydantic>=2->ydata-profiling) (4.7.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\zhiyan\anaconda3
\lib\site-packages (from requests<3,>=2.24.0->ydata-profiling) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\zhiyan\anaconda3\lib\site-pac
kages (from requests<3,>=2.24.0->ydata-profiling) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\zhiyan\anaconda3\lib\si
te-packages (from requests<3,>=2.24.0->ydata-profiling) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\zhiyan\anaconda3\lib\si
te-packages (from requests<3,>=2.24.0->ydata-profiling) (2023.7.22)
Requirement already satisfied: patsy>=0.5.2 in c:\users\zhiyan\anaconda3\lib\site-pac
kages (from statsmodels<1,>=0.13.2->ydata-profiling) (0.5.3)
Requirement already satisfied: colorama in c:\users\zhiyan\anaconda3\lib\site-package
s (from tqdm<5,>=4.48.2->ydata-profiling) (0.4.6)
Requirement already satisfied: six in c:\users\zhiyan\anaconda3\lib\site-packages (fr
om patsy>=0.5.2->statsmodels<1,>=0.13.2->ydata-profiling) (1.16.0)
```

In [91]:
```
from pydantic_settings import BaseSettings
from ydata_profiling import ProfileReport
```

In [16]:
```
ProfileReport(df)
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Overview

Dataset statistics

Number of variables	17
Number of observations	4521
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	600.6 KiB
Average record size in memory	136.0 B

Variable types

Numeric	7
Categorical	6
Boolean	4

Alerts

`contact` is highly overall correlated with `month`	**High correlation**
`month` is highly overall correlated with `contact`	**High correlation**

Out[16]:

In [92]:
```
df.corr(numeric_only=True)
```

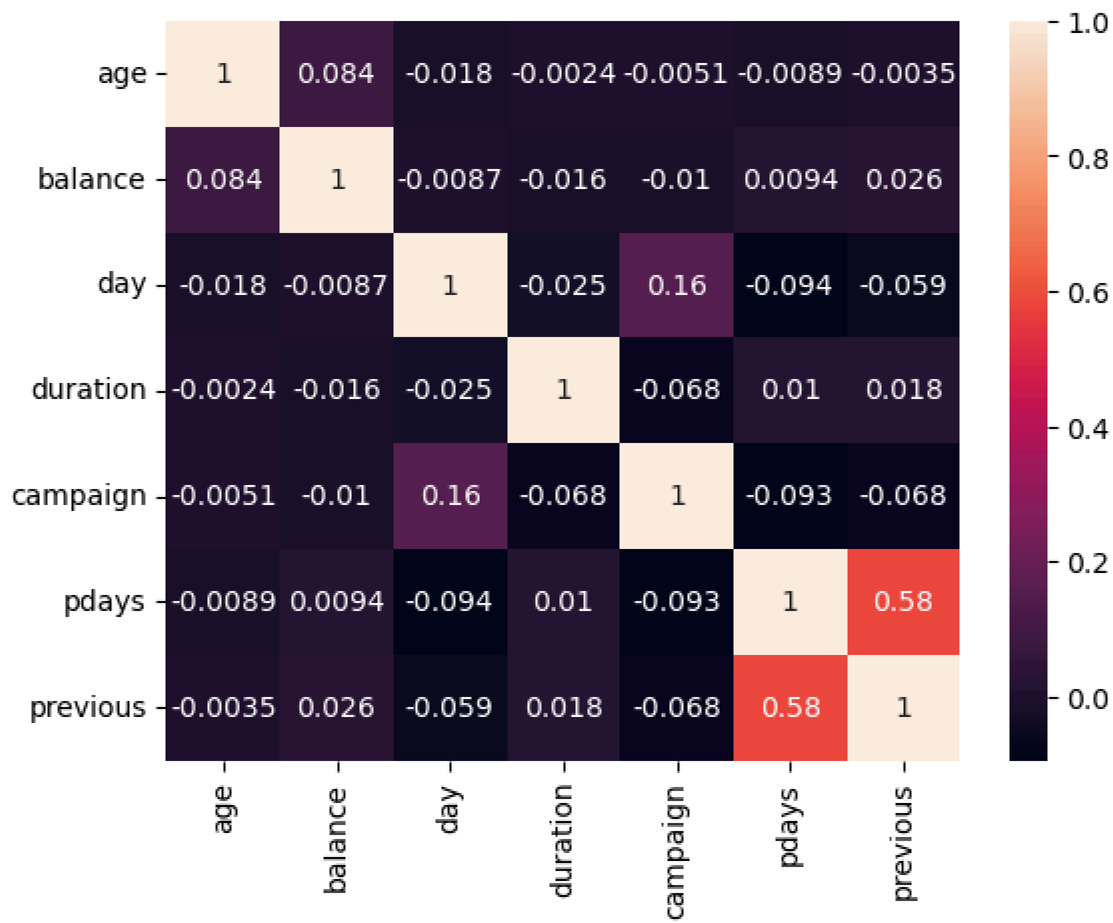
Out[92]:

	age	balance	day	duration	campaign	pdays	previous
age	1.000000	0.083820	-0.017853	-0.002367	-0.005148	-0.008894	-0.003511
balance	0.083820	1.000000	-0.008677	-0.015950	-0.009976	0.009437	0.026196
day	-0.017853	-0.008677	1.000000	-0.024629	0.160706	-0.094352	-0.059114
duration	-0.002367	-0.015950	-0.024629	1.000000	-0.068382	0.010380	0.018080
campaign	-0.005148	-0.009976	0.160706	-0.068382	1.000000	-0.093137	-0.067833
pdays	-0.008894	0.009437	-0.094352	0.010380	-0.093137	1.000000	0.577562
previous	-0.003511	0.026196	-0.059114	0.018080	-0.067833	0.577562	1.000000

In [18]:
```
import seaborn as sns
```

In [13]:
```
sns.heatmap(df.corr(numeric_only=True),annot=True)
```

Out[13]: <Axes: >



In [93]:
```
from sklearn.model_selection import train_test_split
```

In [94]:
```
# Dropping unnecessary columns
ToDrop = ["contact", "day", "month"]
df2 = df.drop(columns = ToDrop)
df2.head()
```

Out[94]:

	age	job	marital	education	default	balance	housing	loan	duration	campaign	pdays
0	30	unemployed	married	primary	no	1787	no	no	79	1	-1
1	33	services	married	secondary	no	4789	yes	yes	220	1	339
2	35	management	single	tertiary	no	1350	yes	no	185	1	330
3	30	management	married	tertiary	no	1476	yes	yes	199	4	-1
4	59	blue-collar	married	secondary	no	0	yes	no	226	1	-1

In [95]:

```
#One-hot-encoding:get_dummies() -> add in more columns to split into inidividual, eg g
#Label encoding: LabelEncoder()

df3 = pd.get_dummies(df2, columns = ['job', 'marital', 'education', 'poutcome'])
df3.head()
```

Out[95]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_marrie
0	30	no	1787	no	no	79	1	-1	0	no	...	Tru
1	33	no	4789	yes	yes	220	1	339	4	no	...	Tru
2	35	no	1350	yes	no	185	1	330	1	no	...	Fals
3	30	no	1476	yes	yes	199	4	-1	0	no	...	Tru
4	59	no	0	yes	no	226	1	-1	0	no	...	Tru

5 rows × 33 columns

In [96]:

```
# Convert 'yes'/'no' to True/False for the specified columns
columns_to_convert = ['default', 'housing', 'loan', 'y']
df3[columns_to_convert] = df3[columns_to_convert].applymap(lambda x: True if x == 'yes
df3.head()
```

Out[96]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_mar
0	30	False	1787	False	False	79	1	-1	0	False	...	
1	33	False	4789	True	True	220	1	339	4	False	...	
2	35	False	1350	True	False	185	1	330	1	False	...	F
3	30	False	1476	True	True	199	4	-1	0	False	...	
4	59	False	0	True	False	226	1	-1	0	False	...	

5 rows × 33 columns

In [19]:

```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 33 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   age                  4521 non-null   int64
 1   default              4521 non-null   bool
 2   balance              4521 non-null   int64
 3   housing              4521 non-null   bool
 4   loan                 4521 non-null   bool
 5   duration             4521 non-null   int64
 6   campaign             4521 non-null   int64
 7   pdays                4521 non-null   int64
 8   previous             4521 non-null   int64
 9   y                    4521 non-null   bool
 10  job_admin.           4521 non-null   bool
 11  job_blue-collar      4521 non-null   bool
 12  job_entrepreneur     4521 non-null   bool
 13  job_housemaid        4521 non-null   bool
 14  job_management       4521 non-null   bool
 15  job_retired          4521 non-null   bool
 16  job_self-employed    4521 non-null   bool
 17  job_services         4521 non-null   bool
 18  job_student          4521 non-null   bool
 19  job_technician       4521 non-null   bool
 20  job_unemployed       4521 non-null   bool
 21  job_unknown          4521 non-null   bool
 22  marital_divorced     4521 non-null   bool
 23  marital_married      4521 non-null   bool
 24  marital_single       4521 non-null   bool
 25  education_primary    4521 non-null   bool
 26  education_secondary  4521 non-null   bool
 27  education_tertiary   4521 non-null   bool
 28  education_unknown    4521 non-null   bool
 29  poutcome_failure     4521 non-null   bool
 30  poutcome_other       4521 non-null   bool
 31  poutcome_success     4521 non-null   bool
 32  poutcome_unknown     4521 non-null   bool
dtypes: bool(27), int64(6)
memory usage: 331.3 KB
```

In [97]:

```
#After clearning the data, now can split data
X=df3.drop("y",axis=1)
Y=df3["y"]
```

In [98]:

```
from sklearn.linear_model import LogisticRegression
```

In [100…

```
lr = LogisticRegression()
from sklearn.model_selection import train_test_split
```

In [101…

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=
```

In [102…

```
lr.fit(X_train, Y_train)
```

```
C:\Users\zhiyan\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: Co
nvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[102]:

```
▾ LogisticRegression
LogisticRegression()
```

In [29]:

```
lrPredict = lr.predict(X_test)
lrPredict
```

Out[29]:

```
array([False,  True, False, ..., False, False, False])
```

In [30]:

```
from sklearn.metrics import accuracy_score, classification_report,confusion_matrix
```

In [31]:

```
lrAccuracy = accuracy_score(Y_test, lrPredict)
lrAccuracy
```

Out[31]:

```
0.8916728076639646
```

In [32]:

```
lrConf=confusion_matrix(Y_test,lrPredict)
lrConf
```

Out[32]:

```
array([[1172,   33],
       [ 114,   38]], dtype=int64)
```

In [33]:

```
print(classification_report(Y_test, lrPredict))
```

```
              precision    recall  f1-score   support

       False       0.91      0.97      0.94      1205
        True       0.54      0.25      0.34       152

    accuracy                           0.89      1357
   macro avg       0.72      0.61      0.64      1357
weighted avg       0.87      0.89      0.87      1357
```

In [ ]:

```
#Look at Precision, recall(Sensitivity), True row only 54%, 25%, 34%, not high as data
#Support data for False is 1205, True data only 152=> data imbalance
#Accuracy is 89%, although is high, but prevision and recall of True are not high, so
```

In [34]:

```
df3.age.value_counts()
```

Out[34]:
```
age
34    231
32    224
31    199
36    188
33    186
     ...
68      2
87      1
81      1
86      1
84      1
Name: count, Length: 67, dtype: int64
```

In [47]:
```
df3.age
```

Out[47]:
```
0       30
1       33
2       35
3       30
4       59
        ..
4516    33
4517    57
4518    57
4519    28
4520    44
Name: age, Length: 4521, dtype: int64
```

In [103…
```
# From actual dataframe, take only those records
# where the age is less than equal to 70
df4 = df3[df3["age"] <= 70]
df4
```

Out[103]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_
0	30	False	1787	False	False	79	1	-1	0	False	...	
1	33	False	4789	True	True	220	1	339	4	False	...	
2	35	False	1350	True	False	185	1	330	1	False	...	
3	30	False	1476	True	True	199	4	-1	0	False	...	
4	59	False	0	True	False	226	1	-1	0	False	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4516	33	False	-333	True	False	329	5	-1	0	False	...	
4517	57	True	-3313	True	True	153	1	-1	0	False	...	
4518	57	False	295	False	False	151	11	-1	0	False	...	
4519	28	False	1137	False	False	129	4	211	3	False	...	
4520	44	False	1136	True	True	345	2	249	7	False	...	

4467 rows × 33 columns

In [104…
```
df4[df4["y"]==True]
```

Out[104]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_r
13	20	False	502	False	False	261	1	-1	0	True	...	
30	68	False	4189	False	False	897	2	-1	0	True	...	
33	32	False	2536	True	False	958	6	-1	0	True	...	
34	49	False	1235	False	False	354	3	-1	0	True	...	
37	32	False	2089	True	False	132	1	-1	0	True	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4494	26	False	668	True	False	576	3	-1	0	True	...	
4503	60	False	362	False	True	816	6	-1	0	True	...	
4504	42	False	1080	True	True	951	3	370	4	True	...	
4505	32	False	620	True	False	1234	3	-1	0	True	...	
4511	46	False	668	True	False	1263	2	-1	0	True	...	

497 rows × 33 columns

In [ ]:
```
Things to do to improve the model's performance:

1. Balanced class
2. Remove the outliers
3. Use stratify while splitting the data
4. Use different relevant models and then compare the performance
5. Use XGboost model

For visualization/presentation:
1. Complete all steps with comments and justifications
2. Interpret the findings
3. Visualize the data before and after cleaning
4. Visualize feature_importance
5. Provide recommendations at the end
```

In [105…
```
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4467 entries, 0 to 4520
Data columns (total 33 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   age                  4467 non-null   int64
 1   default              4467 non-null   bool
 2   balance              4467 non-null   int64
 3   housing              4467 non-null   bool
 4   loan                 4467 non-null   bool
 5   duration             4467 non-null   int64
 6   campaign             4467 non-null   int64
 7   pdays                4467 non-null   int64
 8   previous             4467 non-null   int64
 9   y                    4467 non-null   bool
 10  job_admin.           4467 non-null   bool
 11  job_blue-collar      4467 non-null   bool
 12  job_entrepreneur     4467 non-null   bool
 13  job_housemaid        4467 non-null   bool
 14  job_management       4467 non-null   bool
 15  job_retired          4467 non-null   bool
 16  job_self-employed    4467 non-null   bool
 17  job_services         4467 non-null   bool
 18  job_student          4467 non-null   bool
 19  job_technician       4467 non-null   bool
 20  job_unemployed       4467 non-null   bool
 21  job_unknown          4467 non-null   bool
 22  marital_divorced     4467 non-null   bool
 23  marital_married      4467 non-null   bool
 24  marital_single       4467 non-null   bool
 25  education_primary    4467 non-null   bool
 26  education_secondary  4467 non-null   bool
 27  education_tertiary   4467 non-null   bool
 28  education_unknown    4467 non-null   bool
 29  poutcome_failure     4467 non-null   bool
 30  poutcome_other       4467 non-null   bool
 31  poutcome_success     4467 non-null   bool
 32  poutcome_unknown     4467 non-null   bool
dtypes: bool(27), int64(6)
memory usage: 362.1 KB
```

In [106…

```
X1=df4.drop("y",axis=1)
Y1=df4["y"]
```

In [107…

```
df4[df4["y"]==True]
```

Out[107]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_r
13	20	False	502	False	False	261	1	-1	0	True	...	
30	68	False	4189	False	False	897	2	-1	0	True	...	
33	32	False	2536	True	False	958	6	-1	0	True	...	
34	49	False	1235	False	False	354	3	-1	0	True	...	
37	32	False	2089	True	False	132	1	-1	0	True	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4494	26	False	668	True	False	576	3	-1	0	True	...	
4503	60	False	362	False	True	816	6	-1	0	True	...	
4504	42	False	1080	True	True	951	3	370	4	True	...	
4505	32	False	620	True	False	1234	3	-1	0	True	...	
4511	46	False	668	True	False	1263	2	-1	0	True	...	

497 rows × 33 columns

In [108…

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X1, Y1, test_size=0.3, random_stat
lr.fit(X_train, Y_train)
```

```
C:\Users\zhiyan\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: Co
nvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[108]:

```
▾ LogisticRegression
LogisticRegression()
```

In [109…

```
lrPredict = lr.predict(X_test)
lrPredict
```

Out[109]:

```
array([False, False, False, ..., False, False, False])
```

In [110…

```
from sklearn.metrics import accuracy_score, classification_report,confusion_matrix
lrAccuracy = accuracy_score(Y_test, lrPredict)
lrAccuracy
lrConf=confusion_matrix(Y_test,lrPredict)
lrConf
print(classification_report(Y_test, lrPredict))
```

```
              precision    recall  f1-score   support

       False       0.91      0.98      0.94      1189
        True       0.56      0.21      0.31       152

    accuracy                           0.89      1341
   macro avg       0.73      0.59      0.62      1341
weighted avg       0.87      0.89      0.87      1341
```

In [ ]:
```
# So not much improvement of accuracy due to data outliers, so considering "class bala
#1. take all rows where y=True
#2. tae only 521 rows where y=False
#3. Combine two dataframe use "concat"
#4. Then apply the split and other cleaning
```

In [113…
```
df3["y"].value_counts()
```

Out[113]:
```
y
False    4000
True      521
Name: count, dtype: int64
```

In [114…
```
dfT=df3[df3["y"]==True]
```

In [115…
```
dfT
```

Out[115]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_r
13	20	False	502	False	False	261	1	-1	0	True	...	
30	68	False	4189	False	False	897	2	-1	0	True	...	
33	32	False	2536	True	False	958	6	-1	0	True	...	
34	49	False	1235	False	False	354	3	-1	0	True	...	
36	78	False	229	False	False	97	1	-1	0	True	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4494	26	False	668	True	False	576	3	-1	0	True	...	
4503	60	False	362	False	True	816	6	-1	0	True	...	
4504	42	False	1080	True	True	951	3	370	4	True	...	
4505	32	False	620	True	False	1234	3	-1	0	True	...	
4511	46	False	668	True	False	1263	2	-1	0	True	...	

521 rows × 33 columns

In [116…
```
dfF=df3[df3["y"]==False]
```

In [117…
```
dfF
```

Out[117]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_
0	30	False	1787	False	False	79	1	-1	0	False	...	
1	33	False	4789	True	True	220	1	339	4	False	...	
2	35	False	1350	True	False	185	1	330	1	False	...	
3	30	False	1476	True	True	199	4	-1	0	False	...	
4	59	False	0	True	False	226	1	-1	0	False	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4516	33	False	-333	True	False	329	5	-1	0	False	...	
4517	57	True	-3313	True	True	153	1	-1	0	False	...	
4518	57	False	295	False	False	151	11	-1	0	False	...	
4519	28	False	1137	False	False	129	4	211	3	False	...	
4520	44	False	1136	True	True	345	2	249	7	False	...	

4000 rows × 33 columns

In [119…

```
dfFF=dfF.sample(n=521)
```

In [120…

```
dfFF
```

Out[120]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_
1783	38	False	0	True	False	206	1	-1	0	False	...	
4440	45	False	13117	False	False	42	2	-1	0	False	...	
2910	55	False	96	False	False	340	2	-1	0	False	...	
3175	38	False	156	True	False	544	3	-1	0	False	...	
1665	51	False	2237	True	False	619	1	-1	0	False	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2144	29	False	-478	False	True	528	2	-1	0	False	...	
3002	27	False	3354	True	False	493	5	-1	0	False	...	
4084	45	False	180	True	True	62	2	-1	0	False	...	
3759	58	False	65	False	False	162	1	-1	0	False	...	
1599	25	False	0	True	False	160	1	-1	0	False	...	

521 rows × 33 columns

In [121…

```
data = [dfT, dfFF]
dfconcat = pd.concat(data)
```

In [122…
```
dfconcat
```

Out[122]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_
13	20	False	502	False	False	261	1	-1	0	True	...	
30	68	False	4189	False	False	897	2	-1	0	True	...	
33	32	False	2536	True	False	958	6	-1	0	True	...	
34	49	False	1235	False	False	354	3	-1	0	True	...	
36	78	False	229	False	False	97	1	-1	0	True	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2144	29	False	-478	False	True	528	2	-1	0	False	...	
3002	27	False	3354	True	False	493	5	-1	0	False	...	
4084	45	False	180	True	True	62	2	-1	0	False	...	
3759	58	False	65	False	False	162	1	-1	0	False	...	
1599	25	False	0	True	False	160	1	-1	0	False	...	

1042 rows × 33 columns

In [123…
```
df5 = dfconcat[dfconcat["age"] <= 70]
df5
```

Out[123]:

	age	default	balance	housing	loan	duration	campaign	pdays	previous	y	...	marital_
13	20	False	502	False	False	261	1	-1	0	True	...	
30	68	False	4189	False	False	897	2	-1	0	True	...	
33	32	False	2536	True	False	958	6	-1	0	True	...	
34	49	False	1235	False	False	354	3	-1	0	True	...	
37	32	False	2089	True	False	132	1	-1	0	True	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2144	29	False	-478	False	True	528	2	-1	0	False	...	
3002	27	False	3354	True	False	493	5	-1	0	False	...	
4084	45	False	180	True	True	62	2	-1	0	False	...	
3759	58	False	65	False	False	162	1	-1	0	False	...	
1599	25	False	0	True	False	160	1	-1	0	False	...	

1012 rows × 33 columns

In [131…
```
df5["y"].value_counts()
```

Out[131]:
```
y
False    515
True     497
Name: count, dtype: int64
```

In [132…
```
X2=df5.drop("y",axis=1)
Y2=df5["y"]
```

In [127…
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X2, Y2, test_size=0.3, random_stat
lr.fit(X_train, Y_train)
```

```
C:\Users\zhiyan\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: Co
nvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[127]:
```
▾ LogisticRegression
LogisticRegression()
```

In [133…
```
lrPredict = lr.predict(X_test)
lrPredict
```

Out[133]:
```
array([False, False, False, False,  True, False, False, False, False,
        True, False, False, False, False, False, False, False, False,
       False, False,  True, False,  True, False,  True, False, False,
        True, False,  True,  True,  True,  True, False, False, False,
        True,  True, False,  True,  True,  True,  True, False, False,
       False,  True, False, False,  True, False, False, False, False,
        True,  True, False, False, False, False,  True,  True,  True,
       False,  True, False, False,  True, False,  True,  True,  True,
        True,  True, False, False,  True, False,  True, False, False,
       False, False,  True, False, False,  True,  True, False,  True,
        True,  True, False, False, False, False,  True,  True,  True,
        True, False, False,  True, False,  True,  True,  True, False,
       False,  True, False, False, False, False,  True,  True, False,
       False, False, False,  True, False, False, False,  True, False,
        True, False,  True,  True, False, False, False, False, False,
       False, False, False, False,  True,  True,  True, False,  True,
        True, False,  True,  True, False, False,  True, False, False,
       False,  True, False,  True,  True, False,  True,  True,  True,
       False,  True, False,  True, False, False,  True, False, False,
       False, False, False, False,  True,  True, False,  True,  True,
       False, False,  True, False,  True, False, False, False, False,
        True, False, False,  True,  True,  True,  True,  True, False,
       False,  True, False,  True,  True, False,  True, False, False,
       False, False,  True, False,  True,  True,  True, False, False,
       False,  True,  True, False,  True, False,  True,  True,  True,
       False, False, False, False,  True, False,  True,  True,  True,
       False, False, False,  True,  True,  True,  True, False,  True,
        True,  True,  True,  True, False,  True,  True,  True,  True,
        True,  True, False,  True,  True,  True,  True, False,  True,
       False,  True,  True, False, False,  True,  True,  True, False,
        True,  True, False, False, False, False,  True,  True,  True,
       False,  True,  True, False, False, False,  True, False,  True,
        True,  True, False, False, False,  True,  True,  True, False,
       False,  True,  True, False, False, False,  True])
```

In [136…
```python
from sklearn.metrics import accuracy_score, classification_report,confusion_matrix
lrAccuracy = accuracy_score(Y_test, lrPredict)
lrAccuracy

print(classification_report(Y_test, lrPredict)) # recall are significantly increased
```

```
              precision    recall  f1-score   support

       False       0.78      0.83      0.80       151
        True       0.82      0.76      0.79       153

    accuracy                           0.80       304
   macro avg       0.80      0.80      0.80       304
weighted avg       0.80      0.80      0.80       304
```

In [135…
```python
lrConf=confusion_matrix(Y_test,lrPredict)
lrConf  #26 and 36 data are miss classified.
```

Out[135]:
```
array([[125,  26],
       [ 36, 117]], dtype=int64)
```

In [ ]:
```python
#Below from Parnav => Alternative way to above dfT,dfFF, df5, no need to run below

# Step 1: Take all rows where y is 'True'
df4_yes = df4[df4['y'] == True]
```

```
# Step 2: Take only 521 rows where y is 'False'
df4_no = df4[df4['y'] == False].sample(n=521, random_state=42)

# Step 3: Combine both dataframes
balanced_df4 = pd.concat([df4_yes, df4_no], axis=0)

# Optionally, you might want to shuffle the combined dataframe
balanced_df4 = balanced_df4.sample(frac=1, random_state=42).reset_index(drop=True)

# Print the shape of the balanced DataFrame
print("Shape of balanced DataFrame:", balanced_df4.shape)


X = balanced_df4.drop("y", axis = 1)
Y = balanced_df4["y"]
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=


lrBalanced = LogisticRegression()
lrBalanced.fit(X_train, Y_train)
lr2Predict = lrBalanced.predict(X_test)
print(classification_report(Y_test, lr2Predict))


lrConf2 = confusion_matrix(Y_test, lr2Predict)
lrConf2
```

In [138…

```
#Use "lazypredict" to see which forecast model gives higher accuracy.
!pip install lazypredict

from lazypredict.Supervised import LazyClassifier


lazy = LazyClassifier(verbose=0, ignore_warnings=True, custom_metric=None)
models, predictions = lazy.fit(X_train, X_test, Y_train, Y_test)
models
```

```
Requirement already satisfied: lazypredict in c:\users\zhiyan\anaconda3\lib\site-packages (0.2.12)
Requirement already satisfied: click in c:\users\zhiyan\anaconda3\lib\site-packages (from lazypredict) (8.0.4)
Requirement already satisfied: scikit-learn in c:\users\zhiyan\anaconda3\lib\site-packages (from lazypredict) (1.3.0)
Requirement already satisfied: pandas in c:\users\zhiyan\anaconda3\lib\site-packages (from lazypredict) (2.0.3)
Requirement already satisfied: tqdm in c:\users\zhiyan\anaconda3\lib\site-packages (from lazypredict) (4.65.0)
Requirement already satisfied: joblib in c:\users\zhiyan\anaconda3\lib\site-packages (from lazypredict) (1.1.1)
Requirement already satisfied: lightgbm in c:\users\zhiyan\anaconda3\lib\site-packages (from lazypredict) (4.3.0)
Requirement already satisfied: xgboost in c:\users\zhiyan\anaconda3\lib\site-packages (from lazypredict) (2.0.2)
Requirement already satisfied: colorama in c:\users\zhiyan\anaconda3\lib\site-packages (from click->lazypredict) (0.4.6)
Requirement already satisfied: numpy in c:\users\zhiyan\anaconda3\lib\site-packages (from lightgbm->lazypredict) (1.24.3)
Requirement already satisfied: scipy in c:\users\zhiyan\anaconda3\lib\site-packages (from lightgbm->lazypredict) (1.11.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\zhiyan\anaconda3\lib\site-packages (from pandas->lazypredict) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\zhiyan\anaconda3\lib\site-packages (from pandas->lazypredict) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\zhiyan\anaconda3\lib\site-packages (from pandas->lazypredict) (2023.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\zhiyan\anaconda3\lib\site-packages (from scikit-learn->lazypredict) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\zhiyan\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas->lazypredict) (1.16.0)
```

```
100%|██████████████████████████████████████████
██| 29/29 [00:00<00:00, 34.77it/s]
```

```
[LightGBM] [Info] Number of positive: 344, number of negative: 364
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was
0.000067 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 594
[LightGBM] [Info] Number of data points in the train set: 708, number of used feature
s: 6
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.485876 -> initscore=-0.056512
[LightGBM] [Info] Start training from score -0.056512
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

Out[138]:

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
XGBClassifier	0.81	0.81	0.81	0.81	0.06
LGBMClassifier	0.79	0.79	0.79	0.79	0.05
ExtraTreesClassifier	0.79	0.79	0.79	0.79	0.13
AdaBoostClassifier	0.79	0.79	0.79	0.79	0.08
SVC	0.79	0.79	0.79	0.79	0.02
RandomForestClassifier	0.79	0.79	0.79	0.79	0.17
CalibratedClassifierCV	0.78	0.78	0.78	0.78	0.02
BernoulliNB	0.78	0.78	0.78	0.78	0.01
LinearSVC	0.77	0.77	0.77	0.77	0.02
LogisticRegression	0.77	0.77	0.77	0.77	0.01
NuSVC	0.77	0.77	0.77	0.77	0.02
RidgeClassifierCV	0.76	0.76	0.76	0.76	0.02
BaggingClassifier	0.76	0.76	0.76	0.76	0.04
SGDClassifier	0.76	0.76	0.76	0.76	0.01
RidgeClassifier	0.76	0.76	0.76	0.76	0.01
LinearDiscriminantAnalysis	0.76	0.76	0.76	0.76	0.02
PassiveAggressiveClassifier	0.75	0.75	0.75	0.75	0.01
NearestCentroid	0.75	0.75	0.75	0.75	0.01
KNeighborsClassifier	0.74	0.74	0.74	0.74	0.02
LabelPropagation	0.74	0.74	0.74	0.74	0.02
DecisionTreeClassifier	0.74	0.74	0.74	0.74	0.01
LabelSpreading	0.74	0.74	0.74	0.74	0.01
Perceptron	0.74	0.74	0.74	0.74	0.00
QuadraticDiscriminantAnalysis	0.72	0.72	0.72	0.72	0.02
GaussianNB	0.72	0.72	0.72	0.72	0.02
ExtraTreeClassifier	0.65	0.65	0.65	0.65	0.00
DummyClassifier	0.50	0.50	0.50	0.33	0.01

In [ ]: