Projeto4-Traduzido

March 1, 2023

1 Predicting Customer Churn in Telecom Operators

[1]:
```
# Python Language Version
from platform import python_version
print('Python Language Version Used in This Jupyter Notebook:',␣
 ↪python_version())
```

```
Python Language Version Used in This Jupyter Notebook: 3.9.12
```

[2]:
```
# Imports
#Libraries for saving the Model
import joblib
import pickle

#Data manipulation libraries
import numpy as np
import pandas as pd

#Data visualization libraries
import seaborn as sns
from matplotlib import pyplot as plt

from sklearn.preprocessing import StandardScaler

#Libraries for Machine Learning
import sklearn
from sklearn.model_selection import train_test_split # Split the dataset
from sklearn.model_selection import GridSearchCV # Hyper Parameter Optimization
from sklearn.model_selection import cross_val_score #Model Evaluation
from sklearn.model_selection import RandomizedSearchCV # Hyper Parameter␣
 ↪Optimization

#Libraries with algorithms for Machine Learning
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
```

```
#Libraries calculate the model metrics
from sklearn.metrics import roc_curve, auc, roc_auc_score, confusion_matrix
from sklearn.metrics import accuracy_score

%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

[3]:
```
# Versions of packages used in this notebook jupyter
%reload_ext watermark
%watermark -a "Matheus Francelino Barbosa" --iversions
```

```
Author: Matheus Francelino Barbosa

numpy     : 1.19.5
matplotlib: 3.5.1
pandas    : 1.4.2
joblib    : 1.2.0
seaborn   : 0.11.2
sklearn   : 1.0.2
```

1.1 Loading the Dataset

[4]:
```
# Load the training data
dados_treino = pd.read_csv('dados/projeto4_telecom_treino.csv')
```

[5]:
```
dados_treino.shape
```

[5]:
```
(3333, 21)
```

[6]:
```
# Load the test data
dados_teste = pd.read_csv('dados/projeto4_telecom_teste.csv')
```

[7]:
```
dados_teste.shape
```

[7]:
```
(1667, 21)
```

[8]:
```
dados_treino.columns
```

[8]:
```
Index(['Unnamed: 0', 'state', 'account_length', 'area_code',
       'international_plan', 'voice_mail_plan', 'number_vmail_messages',
       'total_day_minutes', 'total_day_calls', 'total_day_charge',
       'total_eve_minutes', 'total_eve_calls', 'total_eve_charge',
       'total_night_minutes', 'total_night_calls', 'total_night_charge',
       'total_intl_minutes', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'churn'],
```

```
      dtype='object')
```

[9]:
```
dados_treino.sample(10)
```

[9]:

	Unnamed: 0	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	…	total_eve_calls	total_eve_charge	total_night_minutes	total_night_calls
2103	2104	NH	74	area_code_408	no	no	0	298.1	112	50.68	…	100	17.11	214.7	88
2300	2301	FL	106	area_code_510	no	no	0	159.6	94	27.13	…	118	23.53	223.5	65
112	113	AL	98	area_code_408	no	no	0	161.0	117	27.37	…	113	16.23	227.7	113
2311	2312	WA	141	area_code_415	no	no	0	151.5	104	25.76	…	114	20.59	304.2	109
674	675	DE	119	area_code_415	no	no	0	176.8	90	30.06	…	81	19.10	204.6	77
673	674	CO	60	area_code_408	no	no	0	125.1	99	21.27	…	62	21.15	211.3	79
1190	1191	NE	149	area_code_415	no	no	0	156.0	56	26.52	…	116			
1035	1036	VT	119	area_code_510	no	no	0	190.4	74	32.37	…	113			
636	637	KS	121	area_code_408	no	no	0	150.7	105	25.62	…	133			
2453	2454	HI	134	area_code_415	no	yes	38	214.4	93	36.45	…	57			

```
1190              4.76               163.3                104
1035             18.33               161.2                111
636              16.77               169.0                116
2453             17.99               165.0                 79

      total_night_charge  total_intl_minutes  total_intl_calls  \
2103                9.66                 9.7                 4
2300               10.06                 8.8                 3
112                10.25                12.1                 4
2311               13.69                10.8                 2
674                 9.21                 7.5                15
673                 9.51                11.2                 3
1190                7.35                 8.9                 8
1035                7.25                10.0                 1
636                 7.61                 9.2                15
2453                7.43                10.0                 8

      total_intl_charge  number_customer_service_calls  churn
2103               2.62                              2    yes
2300               2.38                              0     no
112                3.27                              4     no
2311               2.92                              1     no
674                2.03                              1     no
673                3.02                              3     no
1190               2.40                              0     no
1035               2.70                              2     no
636                2.48                              1     no
2453               2.70                              1     no

[10 rows x 21 columns]
```

[10]:
```
#Checking Data Types
dados_treino.dtypes
```

[10]:
```
Unnamed: 0                      int64
state                          object
account_length                  int64
area_code                      object
international_plan             object
voice_mail_plan                object
number_vmail_messages           int64
total_day_minutes             float64
total_day_calls                 int64
total_day_charge              float64
total_eve_minutes             float64
total_eve_calls                 int64
total_eve_charge              float64
```

```
total_night_minutes              float64
total_night_calls                  int64
total_night_charge               float64
total_intl_minutes               float64
total_intl_calls                   int64
total_intl_charge                float64
number_customer_service_calls      int64
churn                             object
dtype: object
```

[11]:
```
# Checking only categorical variables
dados_treino.dtypes[dados_treino.dtypes == 'object']
```

[11]:
```
state                object
area_code            object
international_plan   object
voice_mail_plan      object
churn                object
dtype: object
```

[12]:
```
# List of categorical columns
cats = ['state',
        'area_code',
        'international_plan',
        'voice_mail_plan']
```

[13]:
```
#Verifying only numerical variables
dados_treino.dtypes[dados_treino.dtypes != 'object']
```

[13]:
```
Unnamed: 0                       int64
account_length                   int64
number_vmail_messages            int64
total_day_minutes              float64
total_day_calls                  int64
total_day_charge               float64
total_eve_minutes              float64
total_eve_calls                  int64
total_eve_charge               float64
total_night_minutes            float64
total_night_calls                int64
total_night_charge             float64
total_intl_minutes             float64
total_intl_calls                 int64
total_intl_charge              float64
number_customer_service_calls    int64
dtype: object
```

[14]:
```
# List of numerical columns
nums = ['account_length',
        'number_vmail_messages',
        'total_day_minutes',
        'total_day_calls',
        'total_day_charge',
        'total_eve_minutes',
        'total_eve_calls',
        'total_eve_charge',
        'total_night_minutes',
        'total_night_calls',
        'total_night_charge',
        'total_intl_minutes',
        'total_intl_calls',
        'total_intl_charge',
        'number_customer_service_calls']
```

[15]:
```
dados_treino['churn'].value_counts()
```

[15]:
```
no     2850
yes     483
Name: churn, dtype: int64
```

[16]:
```
target = dados_treino['churn']
```

[17]:
```
target.value_counts()
```

[17]:
```
no     2850
yes     483
Name: churn, dtype: int64
```

1.2 Exploring the numerical data

[18]:
```
dados_treino.describe()
```

[18]:
```
       Unnamed: 0  account_length  number_vmail_messages  total_day_minutes  \
count  3333.00000     3333.000000            3333.000000        3333.000000
mean   1667.00000      101.064806               8.099010         179.775098
std     962.29855       39.822106              13.688365          54.467389
min       1.00000        1.000000               0.000000           0.000000
25%     834.00000       74.000000               0.000000         143.700000
50%    1667.00000      101.000000               0.000000         179.400000
75%    2500.00000      127.000000              20.000000         216.400000
max    3333.00000      243.000000              51.000000         350.800000

       total_day_calls  total_day_charge  total_eve_minutes  total_eve_calls  \
count      3333.000000       3333.000000        3333.000000      3333.000000
```

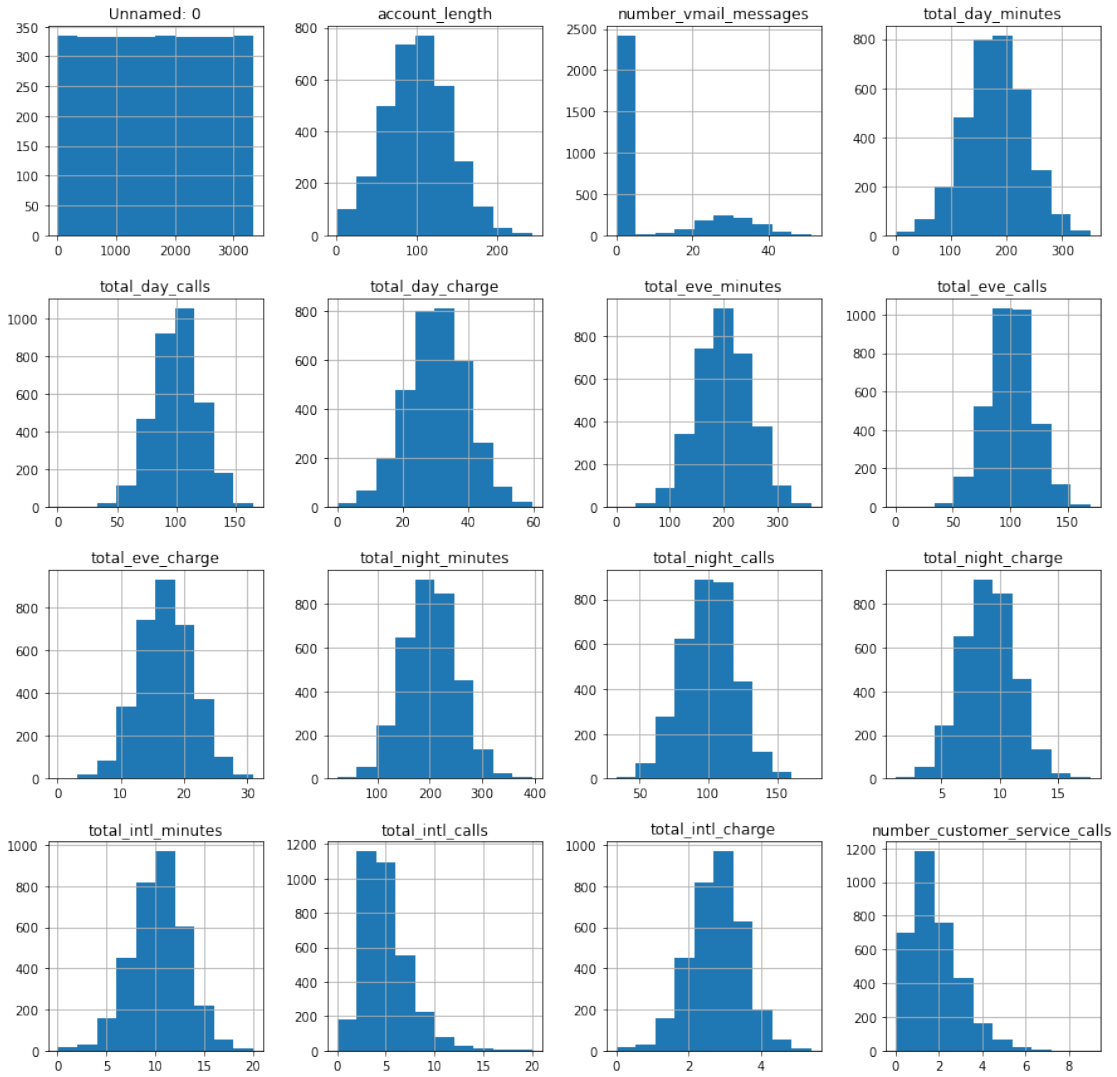
```
mean        100.435644         30.562307         200.980348       100.114311
std          20.069084          9.259435          50.713844        19.922625
min           0.000000          0.000000           0.000000         0.000000
25%          87.000000         24.430000         166.600000        87.000000
50%         101.000000         30.500000         201.400000       100.000000
75%         114.000000         36.790000         235.300000       114.000000
max         165.000000         59.640000         363.700000       170.000000

       total_eve_charge  total_night_minutes  total_night_calls  \
count       3333.000000          3333.000000        3333.000000
mean          17.083540           200.872037         100.107711
std            4.310668            50.573847          19.568609
min            0.000000            23.200000          33.000000
25%           14.160000           167.000000          87.000000
50%           17.120000           201.200000         100.000000
75%           20.000000           235.300000         113.000000
max           30.910000           395.000000         175.000000

       total_night_charge  total_intl_minutes  total_intl_calls  \
count         3333.000000         3333.000000       3333.000000
mean             9.039325           10.237294          4.479448
std              2.275873            2.791840          2.461214
min              1.040000            0.000000          0.000000
25%              7.520000            8.500000          3.000000
50%              9.050000           10.300000          4.000000
75%             10.590000           12.100000          6.000000
max             17.770000           20.000000         20.000000

       total_intl_charge  number_customer_service_calls
count        3333.000000                    3333.000000
mean            2.764581                       1.562856
std             0.753773                       1.315491
min             0.000000                       0.000000
25%             2.300000                       1.000000
50%             2.780000                       1.000000
75%             3.270000                       2.000000
max             5.400000                       9.000000
```

[19]:
```
# Plot
dados_treino.hist(figsize = (15,15), bins = 10)
plt.show()
```



Aparentemente as variaveis seguem uma distribuição normal, exceto pela variavel *number_vmail_messages*

[20]:
```
#Rename the variable target
dados_treino.rename({'churn':'Target'}, axis = 'columns', inplace = True)
```

[21]:
```
dados_treino.columns
```

[21]:
```
Index(['Unnamed: 0', 'state', 'account_length', 'area_code',
       'international_plan', 'voice_mail_plan', 'number_vmail_messages',
       'total_day_minutes', 'total_day_calls', 'total_day_charge',
       'total_eve_minutes', 'total_eve_calls', 'total_eve_charge',
       'total_night_minutes', 'total_night_calls', 'total_night_charge',
       'total_intl_minutes', 'total_intl_calls', 'total_intl_charge',
```

```
       'number_customer_service_calls', 'Target'],
      dtype='object')
```

[22]:
```
# Function for label encoding
# Let's change 'no' to 0 and 'yes' to 1
def encoding_func(x):
    if x == 'no':
        return 0
    return 1
```

[23]:
```
# Apply the function
dados_treino['Target'] = dados_treino['Target'].map(encoding_func)
```

[24]:
```
dados_treino.sample(5)
```

[24]:

	Unnamed: 0	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	…	total_eve_calls	total_eve_charge	total_night_minutes	total_night_calls	total_night_charge	total_intl_minutes	total_intl_calls
2953	2954	GA	136	area_code_415	no	no	0	163.4	83	27.78	…	119	21.19	249.7	90	11.24	9.8	4
2501	2502	IL	101	area_code_415	no	no	0	124.8	66	21.22	…	85	21.86	193.2	115	8.69	13.4	4
407	408	DE	122	area_code_510	no	no	0	157.1	134	26.71	…	122	15.72	197.2	59	8.87	8.5	5
3004	3005	RI	76	area_code_415	no	no	0	171.1	78	29.09	…	83	21.86	91.6	92			
1504	1505	OH	65	area_code_510	no	no	0	153.9	117	26.16	…	122	18.71	280.5	147			

```
3004                4.12                16.2                3
1504               12.62                 8.5                3

      total_intl_charge  number_customer_service_calls  Target
2953               2.65                              7       0
2501               3.62                              0       0
407                2.30                              4       1
3004               4.37                              1       0
1504               2.30                              2       0

[5 rows x 21 columns]
```

[25]:
```
dados_treino['Target'].value_counts()
```

[25]:
```
0    2850
1     483
Name: Target, dtype: int64
```

[26]:
```
target = dados_treino['Target']
```

[27]:
```
# List of numerical columns
nums = ['account_length',
        'number_vmail_messages',
        'total_day_minutes',
        'total_day_calls',
        'total_day_charge',
        'total_eve_minutes',
        'total_eve_calls',
        'total_eve_charge',
        'total_night_minutes',
        'total_night_calls',
        'total_night_charge',
        'total_intl_minutes',
        'total_intl_calls',
        'total_intl_charge',
        'number_customer_service_calls',
        'Target']
```

[28]:
```
# Correlation between numerical variables
dados_treino.corr()
```

[28]:
```
                              Unnamed: 0  account_length  \
Unnamed: 0                      1.000000        0.036667
account_length                  0.036667        1.000000
number_vmail_messages          -0.018086       -0.004628
total_day_minutes              -0.020769        0.006216
total_day_calls                 0.000272        0.038470
```

```
total_day_charge               -0.020769        0.006214
total_eve_minutes               0.013872       -0.006757
total_eve_calls                 0.009149        0.019260
total_eve_charge                0.013875       -0.006745
total_night_minutes             0.011295       -0.008955
total_night_calls               0.000995       -0.013176
total_night_charge              0.011311       -0.008960
total_intl_minutes              0.005822        0.009514
total_intl_calls               -0.011221        0.020661
total_intl_charge               0.005780        0.009546
number_customer_service_calls   0.009665       -0.003796
Target                          0.040232        0.016541

                           number_vmail_messages  total_day_minutes  \
Unnamed: 0                             -0.018086          -0.020769
account_length                         -0.004628           0.006216
number_vmail_messages                   1.000000           0.000778
total_day_minutes                       0.000778           1.000000
total_day_calls                        -0.009548           0.006750
total_day_charge                        0.000776           1.000000
total_eve_minutes                       0.017562           0.007043
total_eve_calls                        -0.005864           0.015769
total_eve_charge                        0.017578           0.007029
total_night_minutes                     0.007681           0.004323
total_night_calls                       0.007123           0.022972
total_night_charge                      0.007663           0.004300
total_intl_minutes                      0.002856          -0.010155
total_intl_calls                        0.013957           0.008033
total_intl_charge                       0.002884          -0.010092
number_customer_service_calls          -0.013263          -0.013423
Target                                 -0.089728           0.205151

                           total_day_calls  total_day_charge  \
Unnamed: 0                        0.000272         -0.020769
account_length                    0.038470          0.006214
number_vmail_messages            -0.009548          0.000776
total_day_minutes                 0.006750          1.000000
total_day_calls                   1.000000          0.006753
total_day_charge                  0.006753          1.000000
total_eve_minutes                -0.021451          0.007050
total_eve_calls                   0.006462          0.015769
total_eve_charge                 -0.021449          0.007036
total_night_minutes               0.022938          0.004324
total_night_calls                -0.019557          0.022972
total_night_charge                0.022927          0.004301
total_intl_minutes                0.021565         -0.010157
total_intl_calls                  0.004574          0.008032
```

```
total_intl_charge                      0.021666         -0.010094
number_customer_service_calls         -0.018942         -0.013427
Target                                 0.018459          0.205151

                              total_eve_minutes  total_eve_calls  \
Unnamed: 0                             0.013872         0.009149
account_length                        -0.006757         0.019260
number_vmail_messages                  0.017562        -0.005864
total_day_minutes                      0.007043         0.015769
total_day_calls                       -0.021451         0.006462
total_day_charge                       0.007050         0.015769
total_eve_minutes                      1.000000        -0.011430
total_eve_calls                       -0.011430         1.000000
total_eve_charge                       1.000000        -0.011423
total_night_minutes                   -0.012584        -0.002093
total_night_calls                      0.007586         0.007710
total_night_charge                    -0.012593        -0.002056
total_intl_minutes                    -0.011035         0.008703
total_intl_calls                       0.002541         0.017434
total_intl_charge                     -0.011067         0.008674
number_customer_service_calls         -0.012985         0.002423
Target                                 0.092796         0.009233

                              total_eve_charge  total_night_minutes  \
Unnamed: 0                            0.013875             0.011295
account_length                       -0.006745            -0.008955
number_vmail_messages                 0.017578             0.007681
total_day_minutes                     0.007029             0.004323
total_day_calls                      -0.021449             0.022938
total_day_charge                      0.007036             0.004324
total_eve_minutes                     1.000000            -0.012584
total_eve_calls                      -0.011423            -0.002093
total_eve_charge                      1.000000            -0.012592
total_night_minutes                  -0.012592             1.000000
total_night_calls                     0.007596             0.011204
total_night_charge                   -0.012601             0.999999
total_intl_minutes                   -0.011043            -0.015207
total_intl_calls                      0.002541            -0.012353
total_intl_charge                    -0.011074            -0.015180
number_customer_service_calls        -0.012987            -0.009288
Target                                0.092786             0.035493

                              total_night_calls  total_night_charge  \
Unnamed: 0                             0.000995            0.011311
account_length                        -0.013176           -0.008960
number_vmail_messages                  0.007123            0.007663
total_day_minutes                      0.022972            0.004300
```

```
total_day_calls                -0.019557            0.022927
total_day_charge                0.022972            0.004301
total_eve_minutes               0.007586           -0.012593
total_eve_calls                 0.007710           -0.002056
total_eve_charge                0.007596           -0.012601
total_night_minutes             0.011204            0.999999
total_night_calls               1.000000            0.011188
total_night_charge              0.011188            1.000000
total_intl_minutes             -0.013605           -0.015214
total_intl_calls                0.000305           -0.012329
total_intl_charge              -0.013630           -0.015186
number_customer_service_calls  -0.012802           -0.009277
Target                          0.006141            0.035496

                           total_intl_minutes  total_intl_calls  \
Unnamed: 0                           0.005822         -0.011221
account_length                       0.009514          0.020661
number_vmail_messages                0.002856          0.013957
total_day_minutes                   -0.010155          0.008033
total_day_calls                      0.021565          0.004574
total_day_charge                    -0.010157          0.008032
total_eve_minutes                   -0.011035          0.002541
total_eve_calls                      0.008703          0.017434
total_eve_charge                    -0.011043          0.002541
total_night_minutes                 -0.015207         -0.012353
total_night_calls                   -0.013605          0.000305
total_night_charge                  -0.015214         -0.012329
total_intl_minutes                   1.000000          0.032304
total_intl_calls                     0.032304          1.000000
total_intl_charge                    0.999993          0.032372
number_customer_service_calls       -0.009640         -0.017561
Target                               0.068239         -0.052844

                           total_intl_charge  \
Unnamed: 0                          0.005780
account_length                      0.009546
number_vmail_messages               0.002884
total_day_minutes                  -0.010092
total_day_calls                     0.021666
total_day_charge                   -0.010094
total_eve_minutes                  -0.011067
total_eve_calls                     0.008674
total_eve_charge                   -0.011074
total_night_minutes                -0.015180
total_night_calls                  -0.013630
total_night_charge                 -0.015186
total_intl_minutes                  0.999993
```

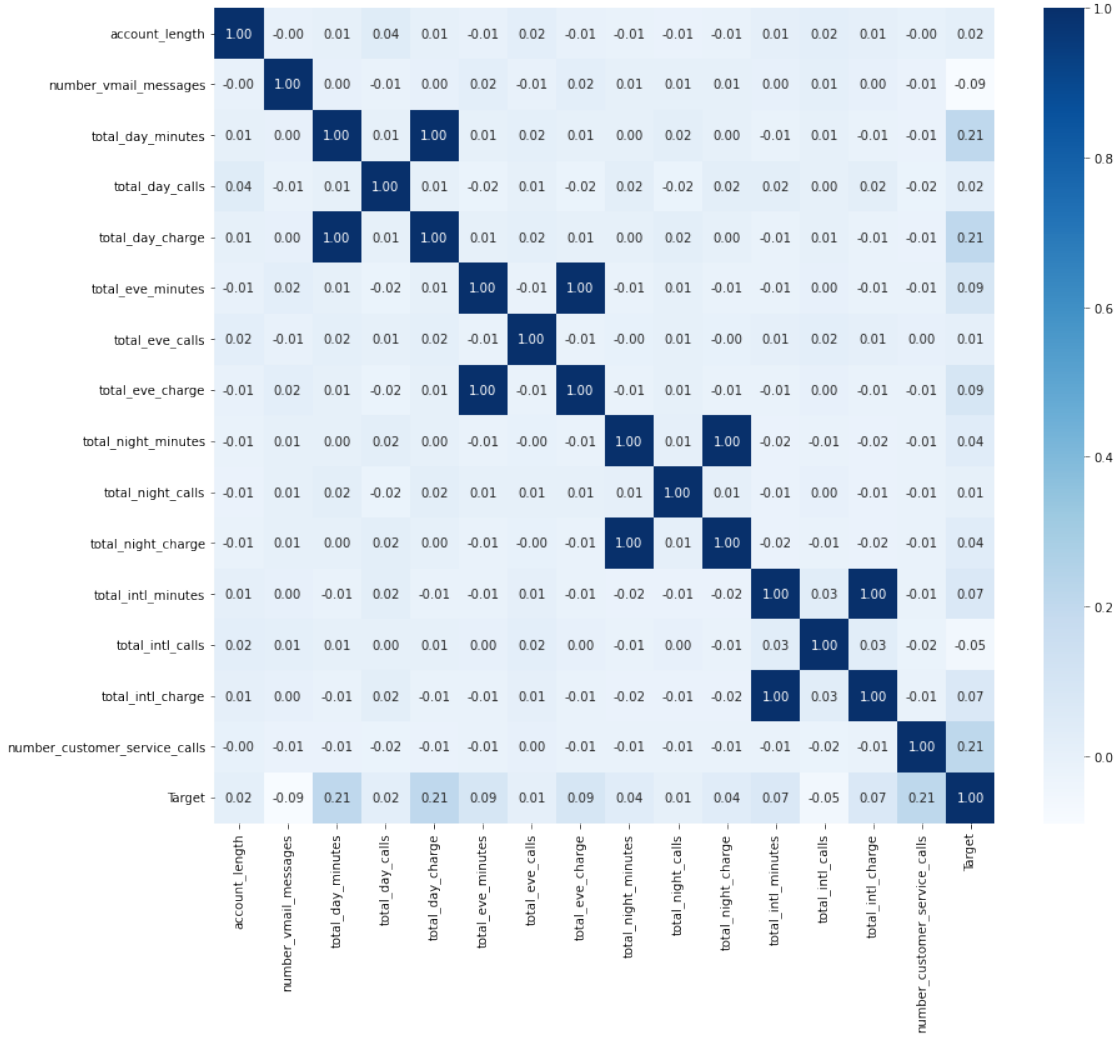
```
total_intl_calls                        0.032372
total_intl_charge                       1.000000
number_customer_service_calls          -0.009675
Target                                  0.068259
```

```
                              number_customer_service_calls    Target
Unnamed: 0                                         0.009665  0.040232
account_length                                    -0.003796  0.016541
number_vmail_messages                             -0.013263 -0.089728
total_day_minutes                                 -0.013423  0.205151
total_day_calls                                   -0.018942  0.018459
total_day_charge                                  -0.013427  0.205151
total_eve_minutes                                 -0.012985  0.092796
total_eve_calls                                    0.002423  0.009233
total_eve_charge                                  -0.012987  0.092786
total_night_minutes                               -0.009288  0.035493
total_night_calls                                 -0.012802  0.006141
total_night_charge                                -0.009277  0.035496
total_intl_minutes                                -0.009640  0.068239
total_intl_calls                                  -0.017561 -0.052844
total_intl_charge                                 -0.009675  0.068259
number_customer_service_calls                      1.000000  0.208750
Target                                             0.208750  1.000000
```

[29]:
```
corr_df = dados_treino[nums].corr()
```

[30]:
```
# Correlation (visual)
plt.figure(figsize = (14, 12))
sns.heatmap(corr_df, cmap = 'Blues', annot = True, fmt = '.2f') #cmap = 'Reds'
```

[30]:
```
<AxesSubplot:>
```

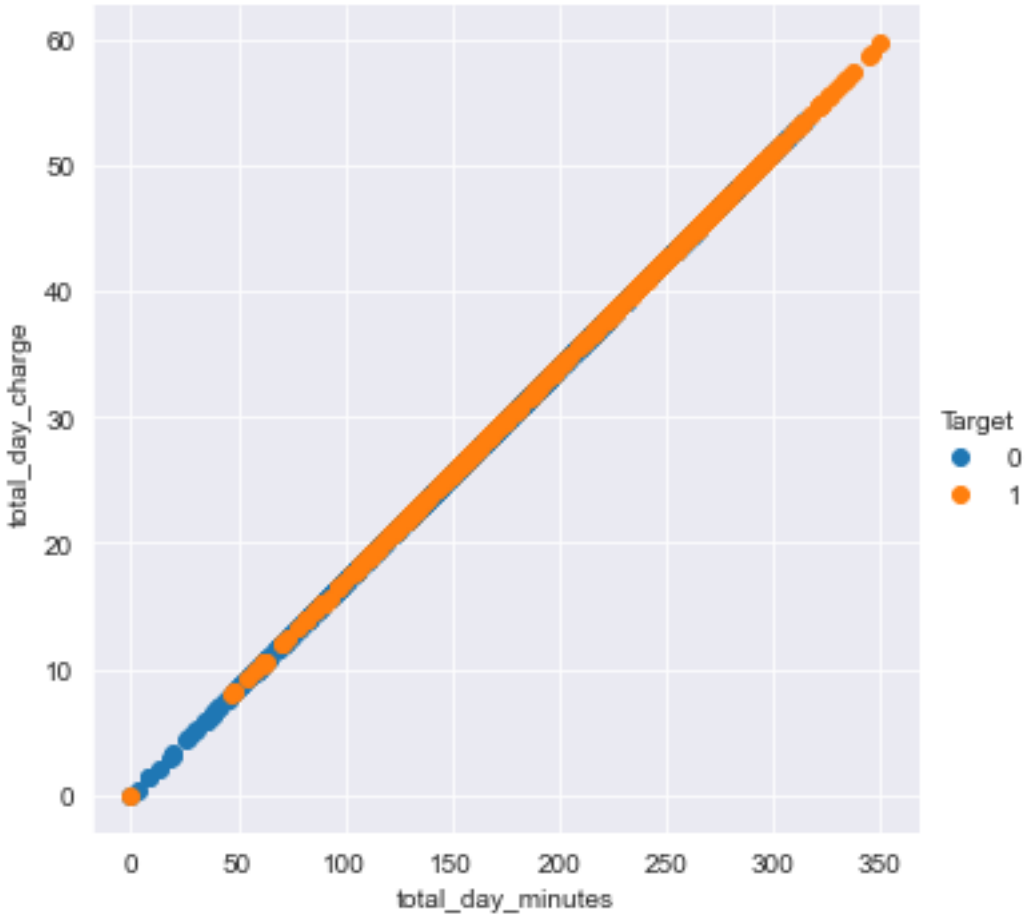


1.2.1 Checking the relationship between attributes

[31]:
```
# Set the background style
sns.set_style('darkgrid')

# Facetgrid
sns.FacetGrid(dados_treino, hue = 'Target', size = 5).map(plt.scatter,␣
 ↪'total_day_minutes', 'total_day_charge').add_legend()
```

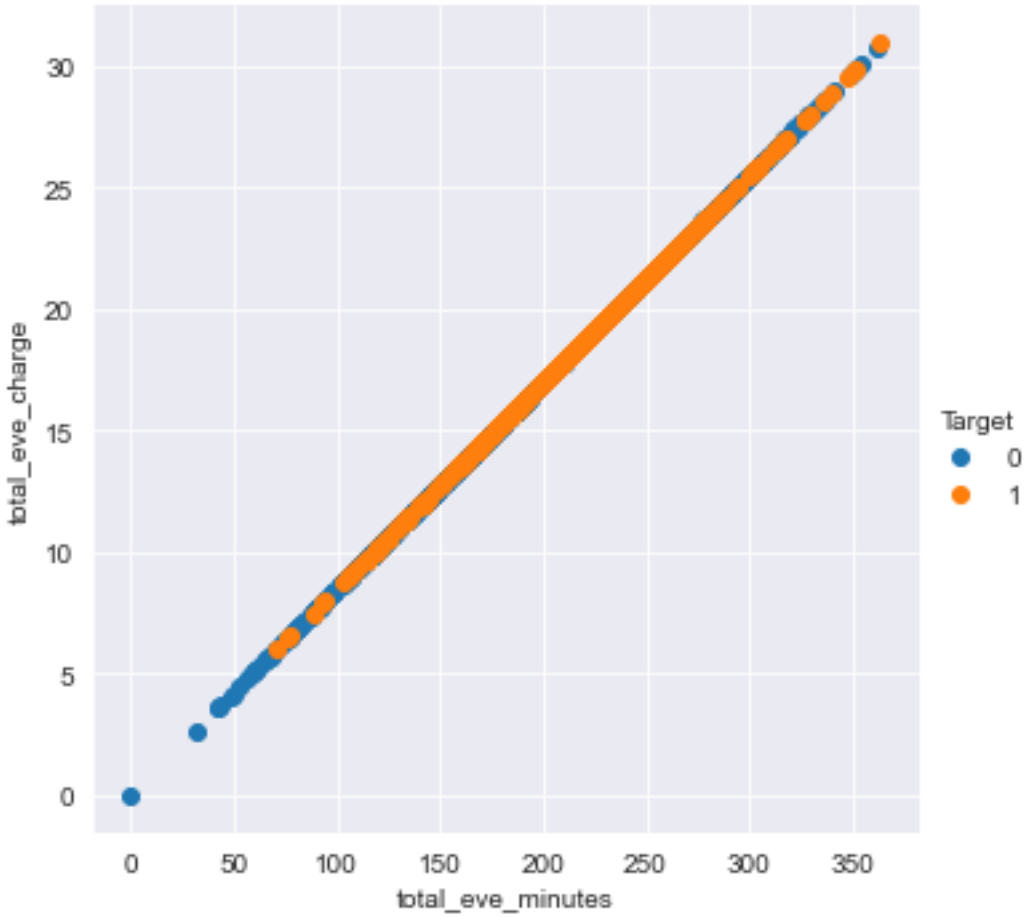
[31]: <seaborn.axisgrid.FacetGrid at 0x20cf6e45b20>



[32]:
```
# Set the background style
sns.set_style('darkgrid')

# Facetgrid
sns.FacetGrid(dados_treino, hue = 'Target', size = 5).map(plt.scatter,␣
↪'total_eve_minutes', 'total_eve_charge').add_legend()
```

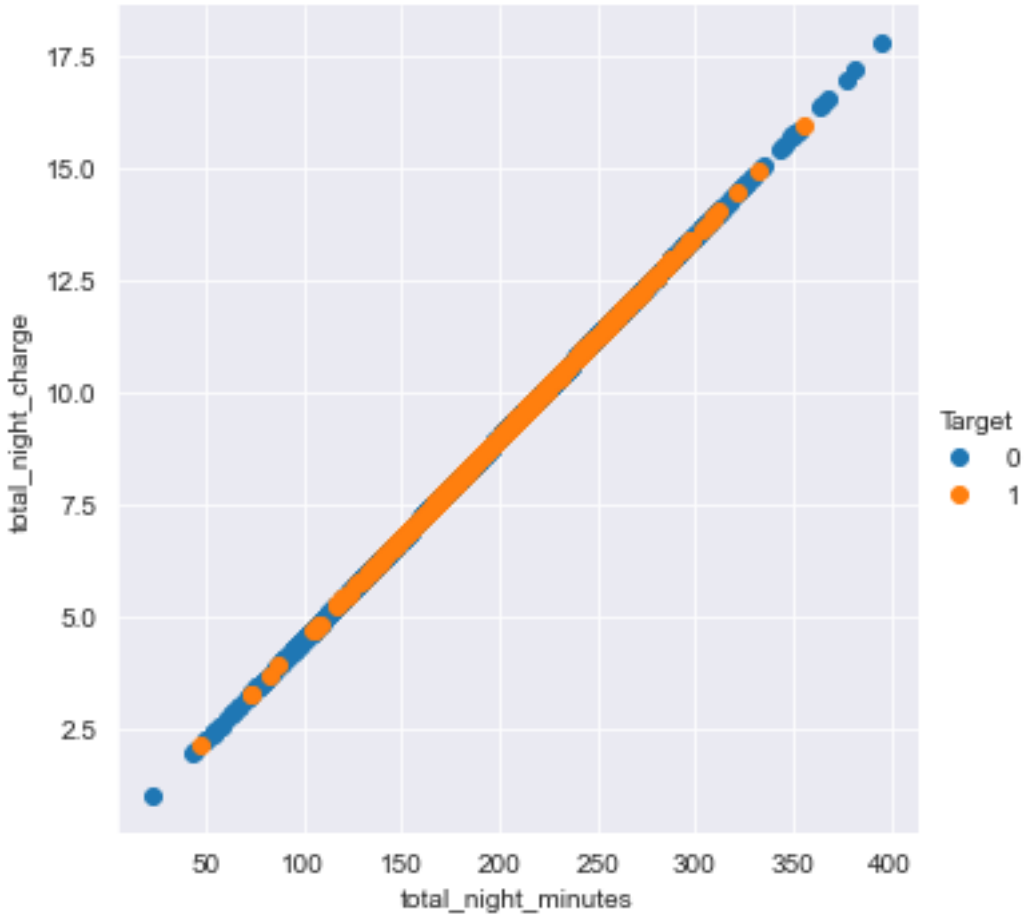
[32]: <seaborn.axisgrid.FacetGrid at 0x20cf6e456d0>



[33]:
```
# Set the background style
sns.set_style('darkgrid')

# Facetgrid
sns.FacetGrid(dados_treino, hue = 'Target', size = 5).map(plt.scatter,␣
 ↪'total_night_minutes', 'total_night_charge').add_legend()
```

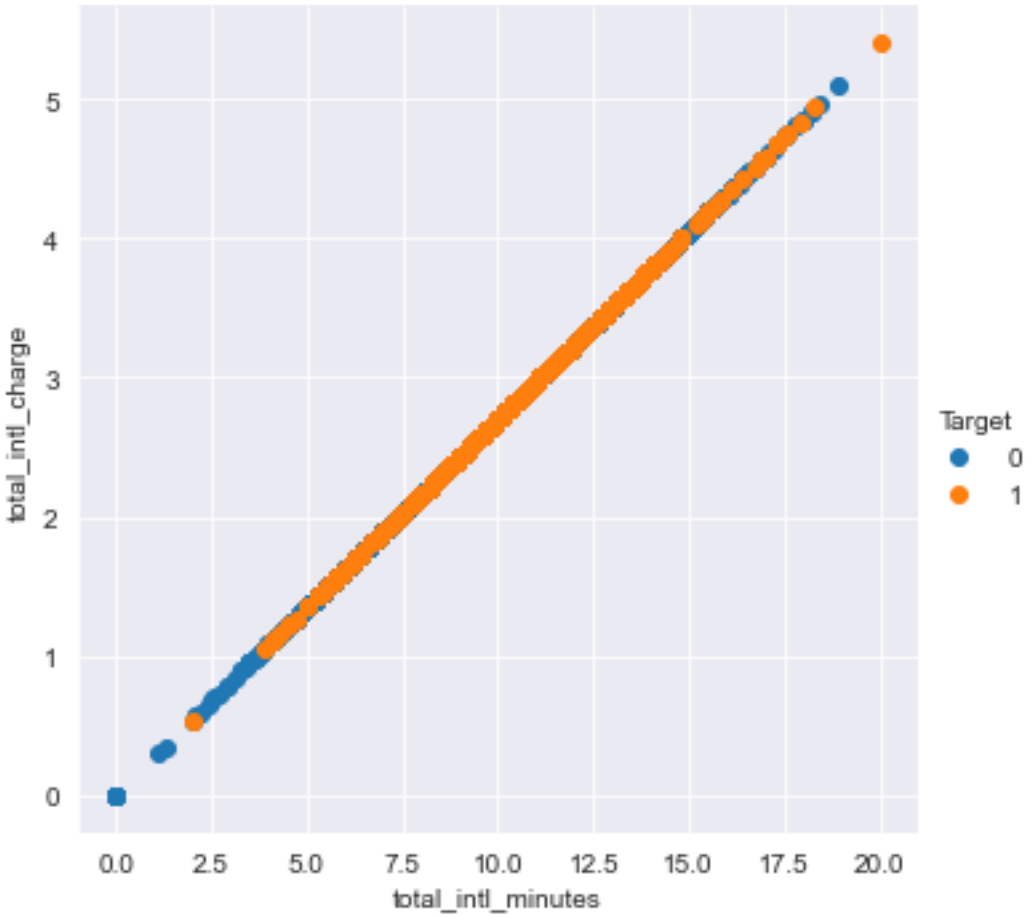
[33]:
```
<seaborn.axisgrid.FacetGrid at 0x20cf64f6e50>
```



[34]:
```
# Set the background style
sns.set_style('darkgrid')

# Facetgrid
sns.FacetGrid(dados_treino, hue = 'Target', size = 5).map(plt.scatter,␣
 ↪'total_intl_minutes', 'total_intl_charge').add_legend()
```

[34]:
```
<seaborn.axisgrid.FacetGrid at 0x20cf6660fa0>
```



In order to avoid strong correlation between the attributes, we can remove the variables total_day_minutes, total_eve_minutes, total_night_minutes and total_intl_minutes

1.3 Exploring the categorical data

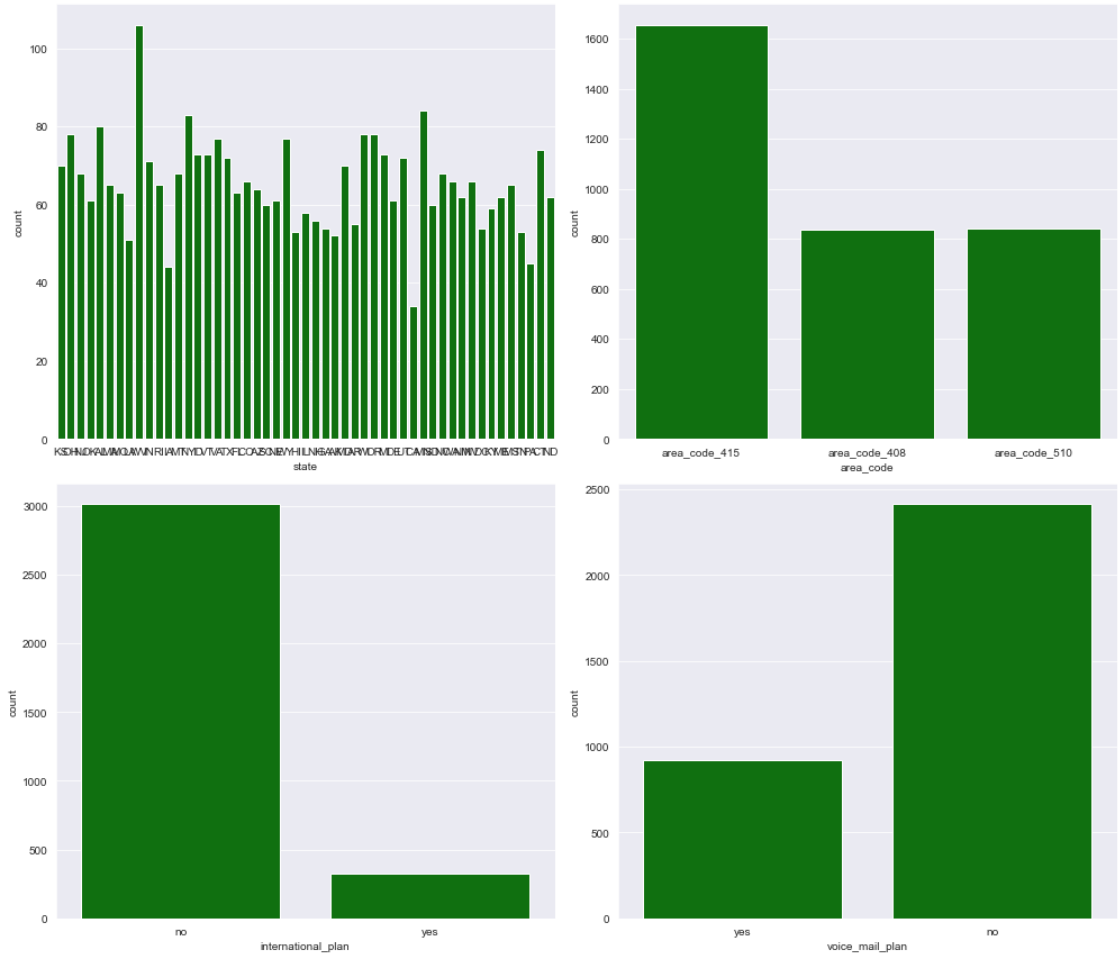
[35]:
```
dados_treino.describe(include = ['object'])
```

[35]:

	state	area_code	international_plan	voice_mail_plan
count	3333	3333	3333	3333
unique	51	3	2	2
top	WV	area_code_415	no	no
freq	106	1655	3010	2411

[36]:
```
plt.figure(figsize = (14, 12))
for i in range(0, len(cats)):
    plt.subplot(2, 2, i+1)
    sns.countplot(x = dados_treino[cats[i]], color = 'green', orient = 'v')
```

```
    plt.tight_layout()
```



1.4 Applying transformations on categorical variables

[37]:
```
# Function for label encoding for international_plan -> 0 = no and 1 = yes
# Apply function
dados_treino['international_plan'] = dados_treino['international_plan'].
  ↪map(encoding_func)
```

[38]:
```
# Function for label encoding for international_plna -> 0 = no and 1 = yes
# Apply the function
dados_treino['voice_mail_plan'] = dados_treino['voice_mail_plan'].
  ↪map(encoding_func)
```

[39]:
```
dados_treino.sample(5)
```

[39]:

	Unnamed: 0	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	…	total_eve_calls	total_eve_charge	total_night_minutes	total_night_calls	total_night_charge	total_intl_minutes	total_intl_calls	total_intl_charge	number_customer_service_calls	Target
3056	3057	IL	131	area_code_510	0	0	0	263.4	123	44.78	…	74	12.91	218.5	101	9.83	10.7	2	2.89	2	0
1488	1489	GA	189	area_code_408	0	0	0	227.4	84	38.66	…	81	14.96	206.1	120	9.27	6.3	4	1.70	2	0
3044	3045	NM	105	area_code_408	0	0	0	146.4	81	24.89	…	80	19.13	230.1	117	10.35	8.5	2	2.30	1	0
3078	3079	AL	107	area_code_408	0	0	0	86.8	95	14.76	…	85	9.19	204.3	87	9.19	13.2	3	3.56	1	0
365	366	CO	154	area_code_415	0	0	0	350.8	75	59.64	…	94	18.40	253.9	100	11.43	10.1	9	2.73	1	1

[5 rows x 21 columns]

[40]:
```
dados_treino.columns
```

[40]:
```
Index(['Unnamed: 0', 'state', 'account_length', 'area_code',
       'international_plan', 'voice_mail_plan', 'number_vmail_messages',
       'total_day_minutes', 'total_day_calls', 'total_day_charge',
       'total_eve_minutes', 'total_eve_calls', 'total_eve_charge',
       'total_night_minutes', 'total_night_calls', 'total_night_charge',
       'total_intl_minutes', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target'],
      dtype='object')
```

[41]:
```
# Checking only categorical variables
dados_treino.dtypes[dados_treino.dtypes == 'object']
```

[41]:
```
state        object
area_code    object
dtype: object
```

[42]:
```
# Checking only the non-categorical variables
dados_treino.dtypes[dados_treino.dtypes != 'object']
```

[42]:
```
Unnamed: 0                      int64
account_length                  int64
international_plan              int64
voice_mail_plan                 int64
number_vmail_messages           int64
total_day_minutes             float64
total_day_calls                 int64
total_day_charge              float64
total_eve_minutes             float64
total_eve_calls                 int64
total_eve_charge              float64
total_night_minutes           float64
total_night_calls               int64
total_night_charge            float64
total_intl_minutes            float64
total_intl_calls                int64
total_intl_charge             float64
number_customer_service_calls   int64
Target                          int64
dtype: object
```

[43]:
```
dados_treino['state'].value_counts()
```

[43]:
```
WV    106
MN     84
NY     83
AL     80
WI     78
OH     78
```

```
OR     78
WY     77
VA     77
CT     74
MI     73
ID     73
VT     73
TX     72
UT     72
IN     71
MD     70
KS     70
NC     68
NJ     68
MT     68
CO     66
NV     66
WA     66
RI     65
MA     65
MS     65
AZ     64
FL     63
MO     63
NM     62
ME     62
ND     62
NE     61
OK     61
DE     61
SC     60
SD     60
KY     59
IL     58
NH     56
AR     55
GA     54
DC     54
HI     53
TN     53
AK     52
LA     51
PA     45
IA     44
CA     34
Name: state, dtype: int64
```

[44]:
```
#Applying one hot enconding to the area code variable
```

[45]:
```
# Applying One-Hot Encoding
for cat in ['area_code']:
    onehots = pd.get_dummies(dados_treino[cat], prefix = cat)
    dados_treino = dados_treino.join(onehots)
```

[46]:
```
dados_treino.columns
```

[46]:
```
Index(['Unnamed: 0', 'state', 'account_length', 'area_code',
       'international_plan', 'voice_mail_plan', 'number_vmail_messages',
       'total_day_minutes', 'total_day_calls', 'total_day_charge',
       'total_eve_minutes', 'total_eve_calls', 'total_eve_charge',
       'total_night_minutes', 'total_night_calls', 'total_night_charge',
       'total_intl_minutes', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target', 'area_code_area_code_408',
       'area_code_area_code_415', 'area_code_area_code_510'],
      dtype='object')
```

[47]:
```
dados_treino.sample(5)
```

[47]:

	Unnamed: 0	state	account_length	area_code	international_plan	\
3100	3101	MA	93	area_code_415	0	
91	92	LA	155	area_code_415	0	
1266	1267	IA	42	area_code_415	0	
273	274	WY	139	area_code_415	0	
816	817	OH	83	area_code_510	0	

	voice_mail_plan	number_vmail_messages	total_day_minutes	\
3100	0	0	173.0	
91	0	0	203.4	
1266	0	0	155.4	
273	0	0	192.8	
816	0	0	227.9	

	total_day_calls	total_day_charge	…	total_night_calls	\
3100	131	29.41	…	66	
91	100	34.58	…	119	
1266	127	26.42	…	128	
273	104	32.78	…	101	
816	78	38.74	…	100	

	total_night_charge	total_intl_minutes	total_intl_calls	\
3100	13.05	10.4	2	
91	8.82	8.9	4	
1266	7.10	9.0	3	
273	9.14	13.0	3	

```
816                 9.53                12.1                   5

      total_intl_charge  number_customer_service_calls  Target  \
3100               2.81                              0       0
91                 2.40                              0       1
1266               2.43                              0       0
273                3.51                              3       0
816                3.27                              1       0

      area_code_area_code_408  area_code_area_code_415  \
3100                        0                        1
91                          0                        1
1266                        0                        1
273                         0                        1
816                         0                        0

      area_code_area_code_510
3100                        0
91                          0
1266                        0
273                         0
816                         1

[5 rows x 24 columns]
```

1.5 Clearing the Data

First we will remove the states column and the area_code column because now we have the one-hot encode

[48]:
```
dados_treino = dados_treino.drop(columns = ['Unnamed: 0',
                         'state',
                         'area_code'])
```

[49]:
```
# Removing the total_day_minutes, total_eve_minutes, total_night_minutes and␣
↪total_intl_minutes columns to avoid correlation
dados_treino = dados_treino.drop(columns = ['total_day_minutes',
                                            'total_eve_minutes',
                                            'total_night_minutes',
                                            'total_intl_minutes'])
```

[50]:
```
dados_treino.sample(5)
```

[50]:
```
      account_length  international_plan  voice_mail_plan  \
1799             132                   0                0
2899              80                   0                1
3198              53                   0                1
650              140                   0                0
```

```
2616             165                    0                 1

      number_vmail_messages  total_day_calls  total_day_charge  \
1799                      0               80             27.74
2899                     36              115             32.35
3198                     32               63             22.30
650                       0               81             40.04
2616                     33              140             18.97

      total_eve_calls  total_eve_charge  total_night_calls  \
1799               90             14.25                 90
2899               78             21.81                145
3198              125             19.33                105
650               130             21.86                111
2616              111             18.13                115

      total_night_charge  total_intl_calls  total_intl_charge  \
1799                3.94                10               1.67
2899                9.67                 4               1.03
3198                8.05                 2               3.46
650                 4.64                 4               3.11
2616               12.04                 3               4.32

      number_customer_service_calls  Target  area_code_area_code_408  \
1799                              1       0                        0
2899                              1       0                        1
3198                              2       0                        0
650                               2       0                        0
2616                              0       0                        0

      area_code_area_code_415  area_code_area_code_510
1799                        0                        1
2899                        0                        0
3198                        1                        0
650                         1                        0
2616                        1                        0
```

[51]:
```
#Rename the variable area_code
dados_treino.rename({'area_code_area_code_408':
 ↪'area_code_408','area_code_area_code_415':
 ↪'area_code_415','area_code_area_code_510':'area_code_510'}, axis =␣
 ↪'columns', inplace = True)
```

[52]:
```
dados_treino.sample(5)
```

[52]:
```
      account_length  international_plan  voice_mail_plan  \
2526              57                   1                0
```

```
746              120                    0                0
1277              68                    0                1
1377             131                    0                1
2838             178                    0                1

      number_vmail_messages  total_day_calls  total_day_charge  \
2526                      0               65             19.55
746                       0               85             25.60
1277                     24              118             29.92
1377                     34              134             26.62
2838                     35               88             29.82

      total_eve_calls  total_eve_charge  total_night_calls  \
2526               96             10.40                 75
746               128             10.12                123
1277              116             23.62                 71
1377               95              6.04                120
2838               65             16.15                 94

      total_night_charge  total_intl_calls  total_intl_charge  \
2526               11.03                 1               1.73
746                10.48                 2               1.73
1277                7.86                 7               3.97
1377               11.78                10               3.62
2838                6.24                 3               2.84

      number_customer_service_calls  Target  area_code_408  area_code_415  \
2526                              0       1              0              0
746                               1       0              0              1
1277                              1       0              0              1
1377                              1       0              0              1
2838                              2       0              0              1

      area_code_510
2526              1
746               0
1277              0
1377              0
2838              0
```

1.5.1 Checking for null and duplicate values

[53]:
```
#Null values
dados_treino[dados_treino.isnull().values]
```

[53]:
```
Empty DataFrame
Columns: [account_length, international_plan, voice_mail_plan,
```

```
number_vmail_messages, total_day_calls, total_day_charge, total_eve_calls,
total_eve_charge, total_night_calls, total_night_charge, total_intl_calls,
total_intl_charge, number_customer_service_calls, Target, area_code_408,
area_code_415, area_code_510]
Index: []
```

[54]:
```
#Duplicate values
dados_treino[dados_treino.duplicated(keep = False)]
```

[54]:
```
Empty DataFrame
Columns: [account_length, international_plan, voice_mail_plan,
number_vmail_messages, total_day_calls, total_day_charge, total_eve_calls,
total_eve_charge, total_night_calls, total_night_charge, total_intl_calls,
total_intl_charge, number_customer_service_calls, Target, area_code_408,
area_code_415, area_code_510]
Index: []
```

1.6 Checking Outliers

[55]:
```
dados_treino.describe()
```

[55]:
```
       account_length  international_plan  voice_mail_plan  \
count     3333.000000         3333.000000      3333.000000
mean       101.064806            0.096910         0.276628
std         39.822106            0.295879         0.447398
min          1.000000            0.000000         0.000000
25%         74.000000            0.000000         0.000000
50%        101.000000            0.000000         0.000000
75%        127.000000            0.000000         1.000000
max        243.000000            1.000000         1.000000

       number_vmail_messages  total_day_calls  total_day_charge  \
count            3333.000000      3333.000000       3333.000000
mean                8.099010       100.435644         30.562307
std                13.688365        20.069084          9.259435
min                 0.000000         0.000000          0.000000
25%                 0.000000        87.000000         24.430000
50%                 0.000000       101.000000         30.500000
75%                20.000000       114.000000         36.790000
max                51.000000       165.000000         59.640000

       total_eve_calls  total_eve_charge  total_night_calls  \
count      3333.000000       3333.000000        3333.000000
mean        100.114311         17.083540         100.107711
std          19.922625          4.310668          19.568609
min           0.000000          0.000000          33.000000
25%          87.000000         14.160000          87.000000
```

```
50%         100.000000         17.120000         100.000000
75%         114.000000         20.000000         113.000000
max         170.000000         30.910000         175.000000

       total_night_charge  total_intl_calls  total_intl_charge  \
count         3333.000000       3333.000000        3333.000000
mean             9.039325          4.479448           2.764581
std              2.275873          2.461214           0.753773
min              1.040000          0.000000           0.000000
25%              7.520000          3.000000           2.300000
50%              9.050000          4.000000           2.780000
75%             10.590000          6.000000           3.270000
max             17.770000         20.000000           5.400000

       number_customer_service_calls       Target  area_code_408  \
count                    3333.000000  3333.000000    3333.000000
mean                        1.562856     0.144914       0.251425
std                         1.315491     0.352067       0.433897
min                         0.000000     0.000000       0.000000
25%                         1.000000     0.000000       0.000000
50%                         1.000000     0.000000       0.000000
75%                         2.000000     0.000000       1.000000
max                         9.000000     1.000000       1.000000

       area_code_415  area_code_510
count    3333.000000    3333.000000
mean        0.496550       0.252025
std         0.500063       0.434241
min         0.000000       0.000000
25%         0.000000       0.000000
50%         0.000000       0.000000
75%         1.000000       1.000000
max         1.000000       1.000000
```

[56]:
```
dados_treino.shape
```

[56]:
```
(3333, 17)
```

[57]:
```
dados_treino.columns
```

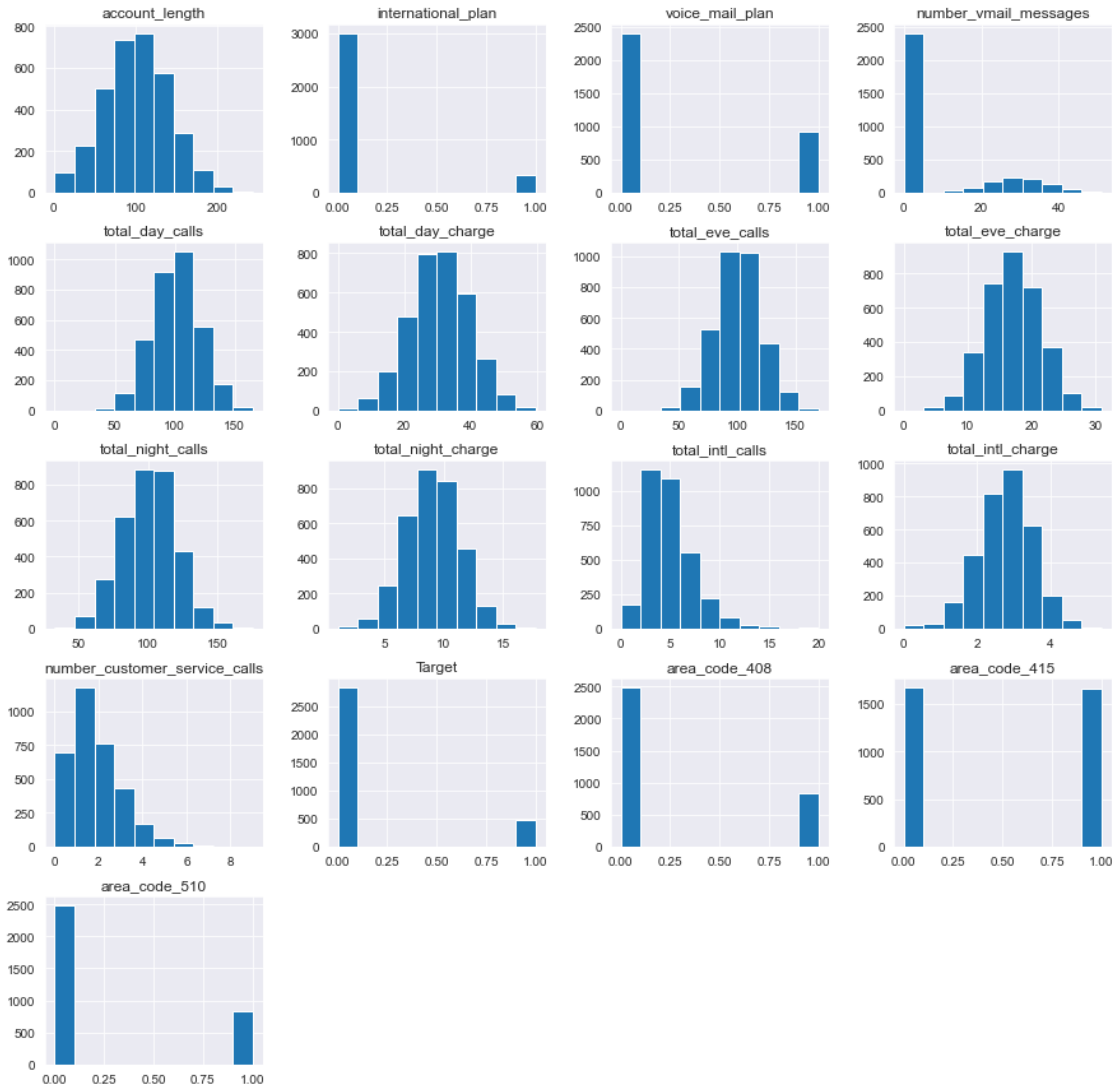
[57]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target', 'area_code_408',
       'area_code_415', 'area_code_510'],
      dtype='object')
```

[58]:
```
var_num = ['account_length',
       'number_vmail_messages', 'total_day_calls',
       'total_day_charge', 'total_eve_calls',
       'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls',
       'total_intl_charge', 'number_customer_service_calls']
```

[59]:
```
var_num
```

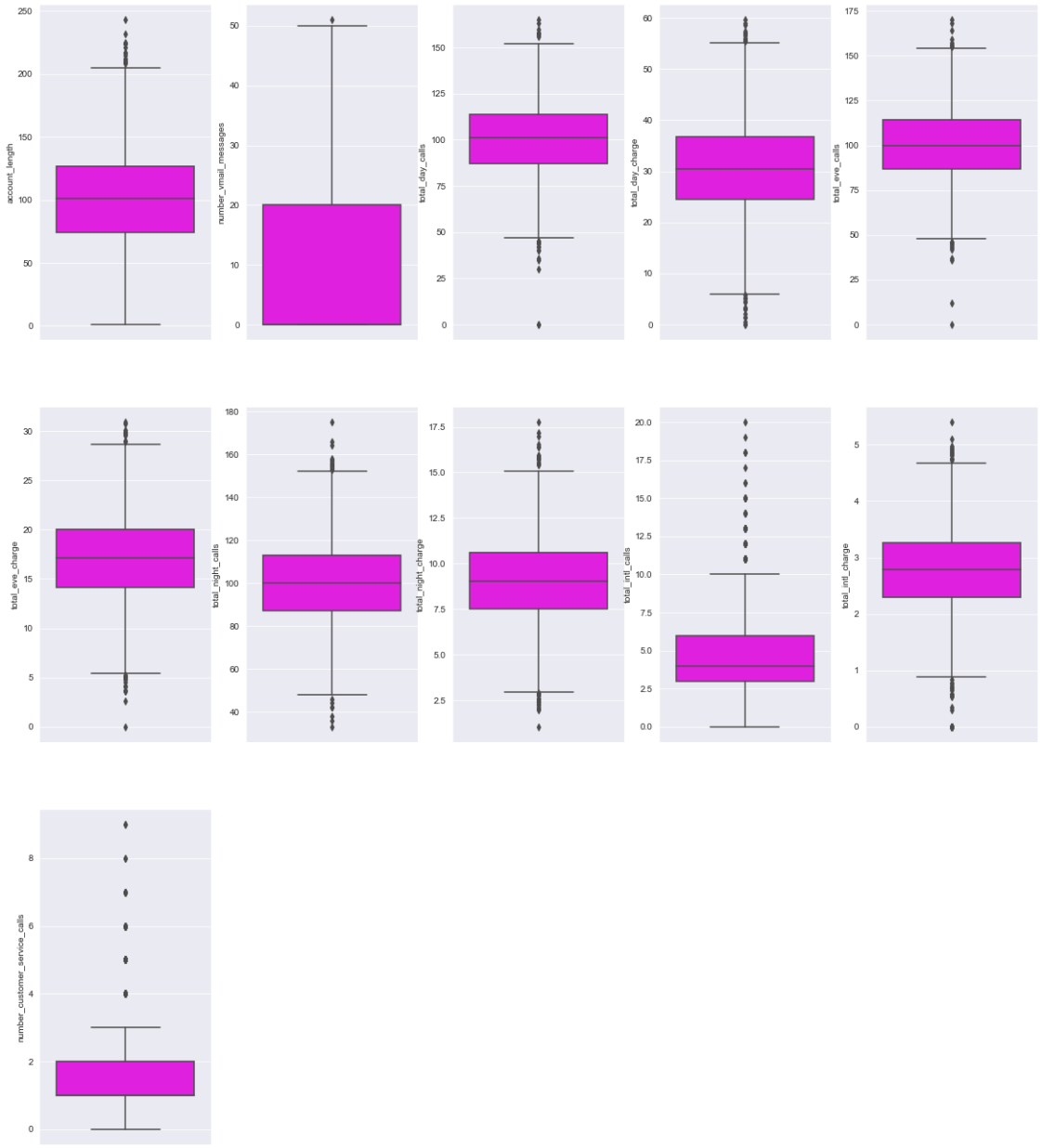
[59]:
```
['account_length',
 'number_vmail_messages',
 'total_day_calls',
 'total_day_charge',
 'total_eve_calls',
 'total_eve_charge',
 'total_night_calls',
 'total_night_charge',
 'total_intl_calls',
 'total_intl_charge',
 'number_customer_service_calls']
```

[60]:
```
# Plot
dados_treino.hist(figsize = (15,15), bins = 10)
plt.show()
```



[61]:
```
plt.figure(figsize = (20, 40))

features = var_num
for i in range(0, len(features)):
    plt.subplot(5, int(len(features)/2), i + 1)
    sns.boxplot(y = dados_treino[features[i]], color = 'magenta', orient = 'v')
    #plt.tight_layout()
```

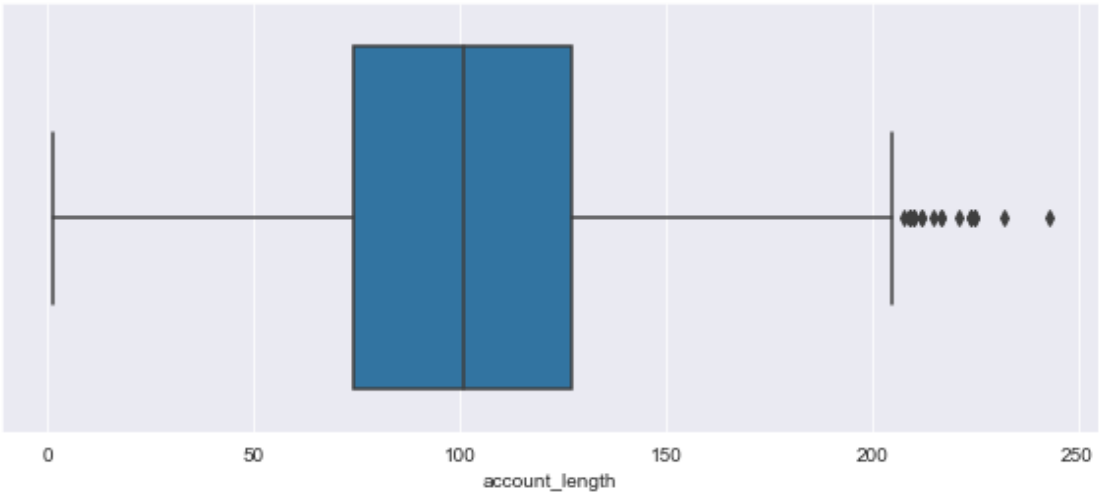


[62]:
```
dados_treino.columns
```

[62]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target', 'area_code_408',
       'area_code_415', 'area_code_510'],
      dtype='object')
```

[63]:
```
# Boxplot
plt.figure(figsize = (10, 4))
sns.boxplot(dados_treino.account_length)
```

[63]:
```
<AxesSubplot:xlabel='account_length'>
```



[64]:
```
#Frequency Counting per Value
dados_treino.account_length.sort_values(ascending = False).head(10)
```

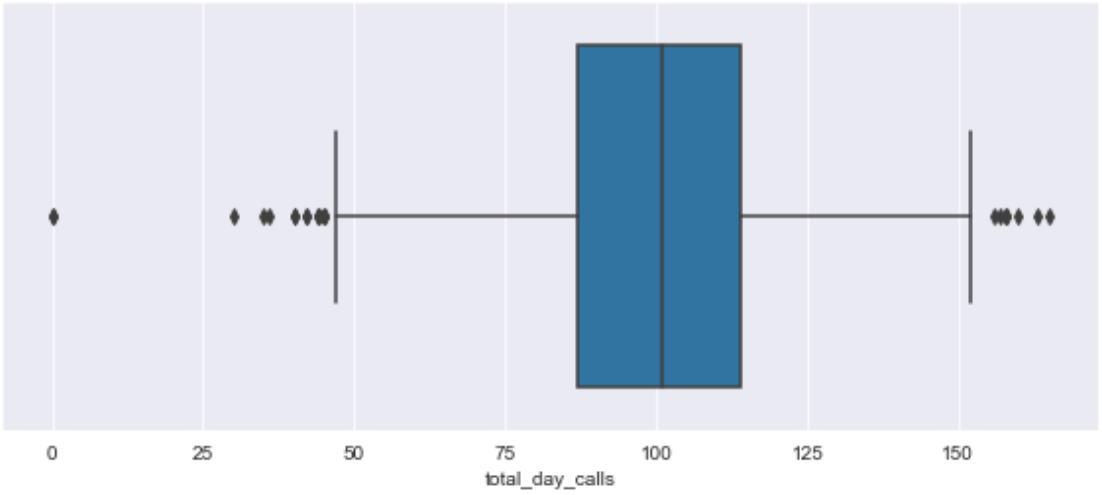
[64]:
```
817     243
1408    232
1751    225
1551    225
1886    224
416     224
3216    221
2277    217
1045    217
238     215
Name: account_length, dtype: int64
```

[65]:
```
# Keep only those records where the value is less than or equal to 220
dados_treino = dados_treino[dados_treino.account_length <= 220]
dados_treino.shape
```

[65]:
```
(3326, 17)
```

[66]:
```
# Boxplot
plt.figure(figsize = (10, 4))
sns.boxplot(dados_treino.total_day_calls)
```

[66]: <AxesSubplot:xlabel='total_day_calls'>



[67]:
```
# Frequency count per value
dados_treino.total_day_calls.sort_values(ascending = True).head(10)
```

[67]:
```
1345     0
1397     0
1144    30
1989    35
692     36
3187    40
740     40
2217    42
1322    42
2884    44
Name: total_day_calls, dtype: int64
```

[68]:
```
# Keep only those records where the value is greater than 40
dados_treino = dados_treino[dados_treino.total_day_calls >= 40]
dados_treino.shape
```

[68]:
```
(3321, 17)
```

[69]:
```
# Frequency count per value
dados_treino.total_day_calls.sort_values(ascending = False).head(10)
```

[69]:
```
1121    165
468     163
1460    160
2392    158
```

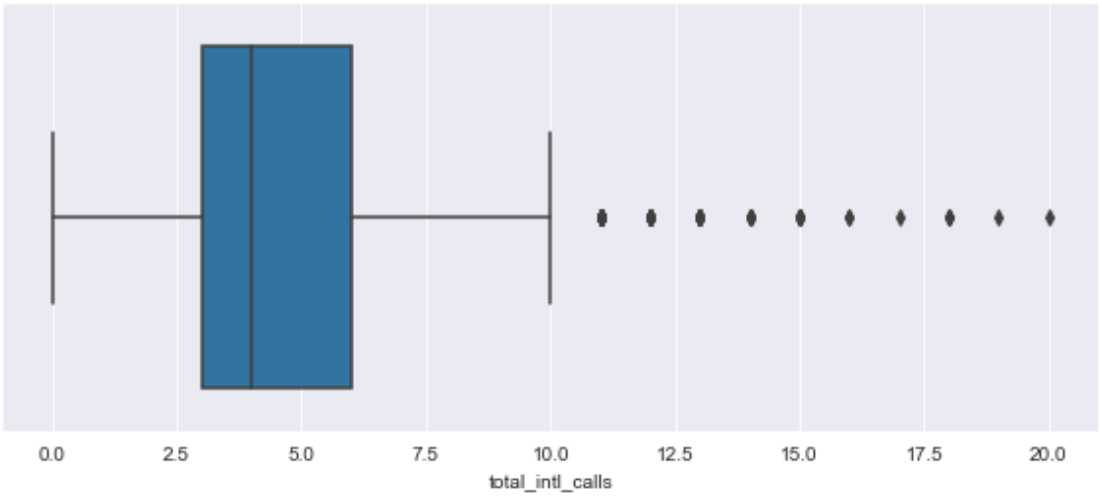
```
1057    158
315     158
2394    157
1869    156
1719    152
164     151
Name: total_day_calls, dtype: int64
```

[70]:
```
# Keep only those records where the value is less than 157
dados_treino = dados_treino[dados_treino.total_day_calls <= 157]
dados_treino.shape
```

[70]:
```
(3315, 17)
```

[71]:
```
# Boxplot
plt.figure(figsize = (10, 4))
sns.boxplot(dados_treino.total_intl_calls)
```

[71]:
```
<AxesSubplot:xlabel='total_intl_calls'>
```



[72]:
```
# Frequency count per value
dados_treino.total_intl_calls.sort_values(ascending = False).head(10)
```

[72]:
```
3291    20
22      19
982     18
377     18
2956    18
3310    17
1567    16
```

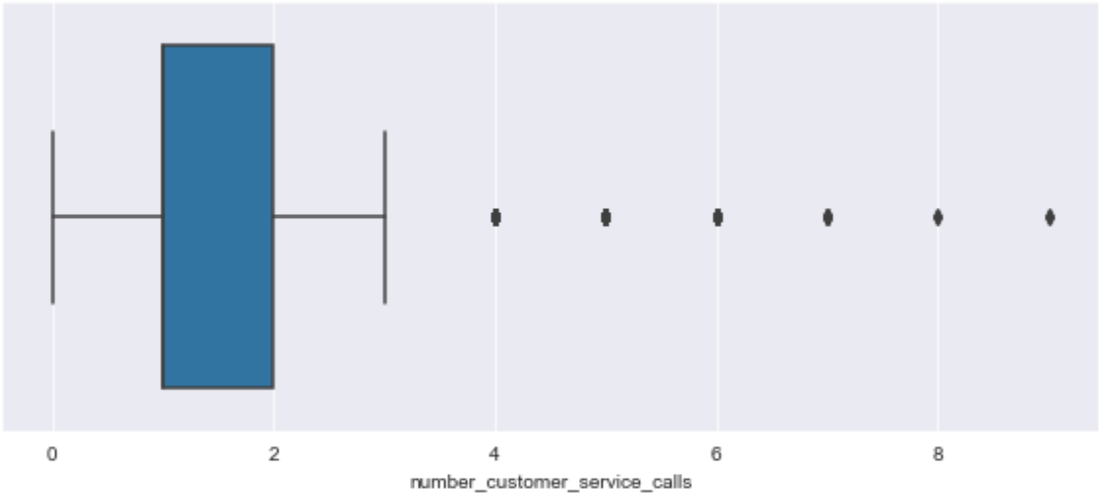
```
2621    16
957     15
1889    15
Name: total_intl_calls, dtype: int64
```

[73]:
```
# Keep only those records where the value is less than 15
dados_treino = dados_treino[dados_treino.total_intl_calls <= 15]
dados_treino.shape
```

[73]:
```
(3307, 17)
```

[74]:
```
# Boxplot
plt.figure(figsize = (10, 4))
sns.boxplot(dados_treino.number_customer_service_calls)
```

[74]:
```
<AxesSubplot:xlabel='number_customer_service_calls'>
```



[75]:
```
# Frequency count per value
dados_treino.number_customer_service_calls.sort_values(ascending = False).
↪head(10)
```

[75]:
```
2380    9
542     9
1912    8
1502    8
902     7
1694    7
2953    7
522     7
2979    7
```

```
1865    7
Name: number_customer_service_calls, dtype: int64
```

[76]:
```
# Keep only those records where the value is less than 6
dados_treino = dados_treino[dados_treino.number_customer_service_calls <= 6]
dados_treino.shape
```

[76]:
```
(3294, 17)
```

[77]:
```
dados_treino.sample(10)
```

[77]:

	account_length	international_plan	voice_mail_plan	\
2190	88	0	1	
1651	64	1	0	
2455	129	0	0	
1905	61	0	0	
1731	70	0	0	
1869	101	0	0	
2480	105	1	0	
2330	69	0	0	
1867	142	1	1	
2541	73	0	0	

	number_vmail_messages	total_day_calls	total_day_charge	\
2190	27	106	15.88	
1651	0	134	38.30	
2455	0	145	17.24	
1905	0	118	33.61	
1731	0	95	39.58	
1869	0	156	40.63	
2480	0	106	25.50	
2330	0	101	23.02	
1867	25	109	32.49	
2541	0	121	16.13	

	total_eve_calls	total_eve_charge	total_night_calls	\
2190	92	21.42	104	
1651	87	9.20	132	
2455	116	21.17	107	
1905	96	12.94	93	
1731	111	25.79	104	
1869	106	23.21	93	
2480	123	24.97	65	
2330	124	20.24	102	
1867	120	12.72	60	
2541	83	21.52	86	

```
      total_night_charge  total_intl_calls  total_intl_charge  \
2190                8.50                 1               2.94
1651                6.28                 9               4.67
2455                7.09                 6               1.92
1905                9.95                 3               1.89
1731               11.50                 7               3.48
1869               12.52                 8               3.65
2480               11.28                 7               2.78
2330                8.80                 2               2.86
1867               10.25                 3               2.65
2541                7.88                 2               3.83

      number_customer_service_calls  Target  area_code_408  area_code_415  \
2190                              1       0              1              0
1651                              1       1              0              1
2455                              1       0              1              0
1905                              2       0              0              1
1731                              0       1              0              1
1869                              1       1              1              0
2480                              3       0              0              1
2330                              1       0              0              0
1867                              0       0              1              0
2541                              2       0              1              0

      area_code_510
2190              0
1651              0
2455              0
1905              0
1731              0
1869              0
2480              0
2330              1
1867              0
2541              0
```

[78]:
```
dados_treino['account_length'].describe()
```

[78]:
```
count    3294.000000
mean      100.792350
std        39.535106
min         1.000000
25%        74.000000
50%       101.000000
75%       127.000000
max       217.000000
Name: account_length, dtype: float64
```

[79]:
```
dados_treino.columns
```

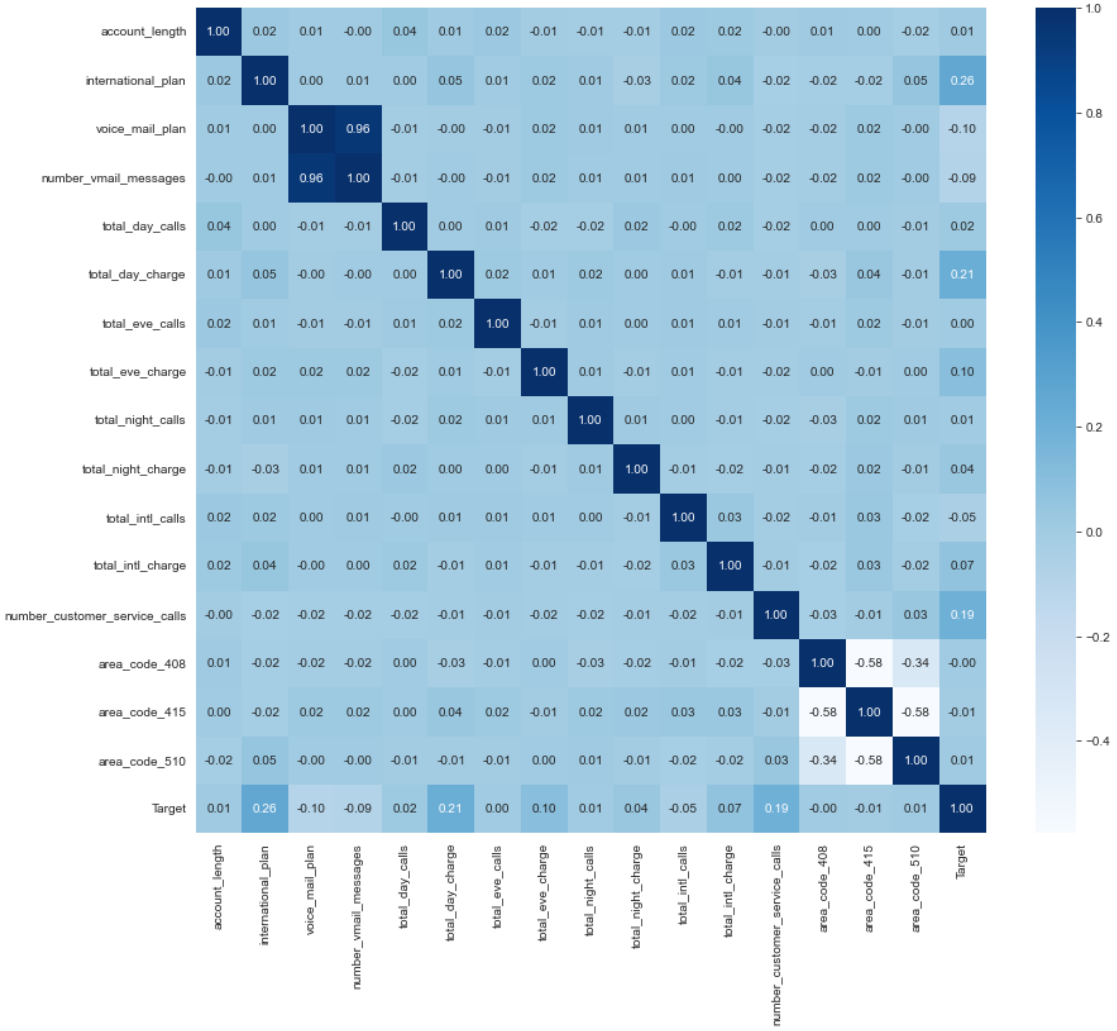
[79]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target', 'area_code_408',
       'area_code_415', 'area_code_510'],
      dtype='object')
```

[80]:
```
num = ['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'area_code_408',
       'area_code_415', 'area_code_510', 'Target']
```

[81]:
```
corr_df2 = dados_treino[num].corr()
```

[82]:
```
# Correlation (visual)
plt.figure(figsize = (14, 12))
sns.heatmap(corr_df2, cmap = 'Blues', annot = True, fmt = '.2f') #cmap = 'Reds'
```

[82]:
```
<AxesSubplot:>
```



[83]:
```
dados_treino.corr()
```

[83]:

	account_length	international_plan	\
account_length	1.000000	0.022469	
international_plan	0.022469	1.000000	
voice_mail_plan	0.005119	0.003437	
number_vmail_messages	-0.001490	0.007717	
total_day_calls	0.043699	0.002333	
total_day_charge	0.008473	0.047156	
total_eve_calls	0.023748	0.005822	
total_eve_charge	-0.007020	0.018707	
total_night_calls	-0.013751	0.013694	
total_night_charge	-0.010142	-0.029781	
total_intl_calls	0.024280	0.017244	
total_intl_charge	0.017324	0.044649	

```
number_customer_service_calls       -0.004444           -0.024543
Target                               0.013276            0.258917
area_code_408                        0.014286           -0.022300
area_code_415                        0.002757           -0.020162
area_code_510                       -0.017443            0.045484

                            voice_mail_plan  number_vmail_messages  \
account_length                     0.005119              -0.001490
international_plan                 0.003437               0.007717
voice_mail_plan                    1.000000               0.956830
number_vmail_messages              0.956830               1.000000
total_day_calls                   -0.014499              -0.013576
total_day_charge                  -0.004830              -0.001621
total_eve_calls                   -0.006888              -0.006360
total_eve_charge                   0.020185               0.016445
total_night_calls                  0.014467               0.005350
total_night_charge                 0.005612               0.007387
total_intl_calls                   0.004031               0.010611
total_intl_charge                 -0.003049               0.001461
number_customer_service_calls     -0.020606              -0.017153
Target                            -0.100878              -0.087185
area_code_408                     -0.015637              -0.015360
area_code_415                      0.016043               0.016586
area_code_510                     -0.002851              -0.003753

                            total_day_calls  total_day_charge  \
account_length                     0.043699          0.008473
international_plan                 0.002333          0.047156
voice_mail_plan                   -0.014499         -0.004830
number_vmail_messages             -0.013576         -0.001621
total_day_calls                    1.000000          0.002196
total_day_charge                   0.002196          1.000000
total_eve_calls                    0.012602          0.018643
total_eve_charge                  -0.021370          0.008022
total_night_calls                 -0.017820          0.020726
total_night_charge                 0.021509          0.003368
total_intl_calls                  -0.003704          0.013177
total_intl_charge                  0.015384         -0.011417
number_customer_service_calls     -0.018914         -0.011379
Target                             0.018737          0.212996
area_code_408                      0.003801         -0.031176
area_code_415                      0.003624          0.035240
area_code_510                     -0.007968         -0.009429

                            total_eve_calls  total_eve_charge  \
account_length                     0.023748         -0.007020
international_plan                 0.005822          0.018707
```

voice_mail_plan	-0.006888	0.020185
number_vmail_messages	-0.006360	0.016445
total_day_calls	0.012602	-0.021370
total_day_charge	0.018643	0.008022
total_eve_calls	1.000000	-0.011096
total_eve_charge	-0.011096	1.000000
total_night_calls	0.009471	0.009569
total_night_charge	0.001234	-0.013562
total_intl_calls	0.013155	0.006560
total_intl_charge	0.007736	-0.010777
number_customer_service_calls	-0.007024	-0.015739
Target	0.003473	0.095806
area_code_408	-0.014245	0.002903
area_code_415	0.020488	-0.005519
area_code_510	-0.009358	0.003454

	total_night_calls	total_night_charge	\
account_length	-0.013751	-0.010142	
international_plan	0.013694	-0.029781	
voice_mail_plan	0.014467	0.005612	
number_vmail_messages	0.005350	0.007387	
total_day_calls	-0.017820	0.021509	
total_day_charge	0.020726	0.003368	
total_eve_calls	0.009471	0.001234	
total_eve_charge	0.009569	-0.013562	
total_night_calls	1.000000	0.008542	
total_night_charge	0.008542	1.000000	
total_intl_calls	0.000048	-0.013826	
total_intl_charge	-0.013058	-0.016392	
number_customer_service_calls	-0.019779	-0.013118	
Target	0.006737	0.036237	
area_code_408	-0.031712	-0.016370	
area_code_415	0.016297	0.021161	
area_code_510	0.012913	-0.008011	

	total_intl_calls	total_intl_charge	\
account_length	0.024280	0.017324	
international_plan	0.017244	0.044649	
voice_mail_plan	0.004031	-0.003049	
number_vmail_messages	0.010611	0.001461	
total_day_calls	-0.003704	0.015384	
total_day_charge	0.013177	-0.011417	
total_eve_calls	0.013155	0.007736	
total_eve_charge	0.006560	-0.010777	
total_night_calls	0.000048	-0.013058	
total_night_charge	-0.013826	-0.016392	
total_intl_calls	1.000000	0.029032	

```
total_intl_charge                0.029032          1.000000
number_customer_service_calls   -0.015928         -0.006779
Target                          -0.053059          0.069386
area_code_408                   -0.008683         -0.022479
area_code_415                    0.029191          0.033040
area_code_510                   -0.024932         -0.015584

                              number_customer_service_calls    Target  \
account_length                                    -0.004444  0.013276
international_plan                                -0.024543  0.258917
voice_mail_plan                                   -0.020606 -0.100878
number_vmail_messages                             -0.017153 -0.087185
total_day_calls                                   -0.018914  0.018737
total_day_charge                                  -0.011379  0.212996
total_eve_calls                                   -0.007024  0.003473
total_eve_charge                                  -0.015739  0.095806
total_night_calls                                 -0.019779  0.006737
total_night_charge                                -0.013118  0.036237
total_intl_calls                                  -0.015928 -0.053059
total_intl_charge                                 -0.006779  0.069386
number_customer_service_calls                      1.000000  0.193816
Target                                             0.193816  1.000000
area_code_408                                     -0.026414 -0.000284
area_code_415                                     -0.006750 -0.007444
area_code_510                                      0.034153  0.008853

                              area_code_408  area_code_415  area_code_510
account_length                     0.014286       0.002757      -0.017443
international_plan                -0.022300      -0.020162       0.045484
voice_mail_plan                   -0.015637       0.016043      -0.002851
number_vmail_messages             -0.015360       0.016586      -0.003753
total_day_calls                    0.003801       0.003624      -0.007968
total_day_charge                  -0.031176       0.035240      -0.009429
total_eve_calls                   -0.014245       0.020488      -0.009358
total_eve_charge                   0.002903      -0.005519       0.003454
total_night_calls                 -0.031712       0.016297       0.012913
total_night_charge                -0.016370       0.021161      -0.008011
total_intl_calls                  -0.008683       0.029191      -0.024932
total_intl_charge                 -0.022479       0.033040      -0.015584
number_customer_service_calls     -0.026414      -0.006750       0.034153
Target                            -0.000284      -0.007444       0.008853
area_code_408                      1.000000      -0.575247      -0.336579
area_code_415                     -0.575247       1.000000      -0.576639
area_code_510                     -0.336579      -0.576639       1.000000
```

[84]:
```
dados_treino.describe()
```

[84]:

	account_length	international_plan	voice_mail_plan	\
count	3294.000000	3294.000000	3294.000000	
mean	100.792350	0.097146	0.276563	
std	39.535106	0.296202	0.447367	
min	1.000000	0.000000	0.000000	
25%	74.000000	0.000000	0.000000	
50%	101.000000	0.000000	0.000000	
75%	127.000000	0.000000	1.000000	
max	217.000000	1.000000	1.000000	

	number_vmail_messages	total_day_calls	total_day_charge	\
count	3294.000000	3294.000000	3294.000000	
mean	8.090771	100.413175	30.621305	
std	13.678046	19.682653	9.238822	
min	0.000000	40.000000	0.440000	
25%	0.000000	87.000000	24.452500	
50%	0.000000	101.000000	30.560000	
75%	19.750000	114.000000	36.860000	
max	51.000000	157.000000	59.640000	

	total_eve_calls	total_eve_charge	total_night_calls	\
count	3294.000000	3294.000000	3294.000000	
mean	100.043109	17.082486	100.117183	
std	19.933703	4.309417	19.577862	
min	0.000000	0.000000	33.000000	
25%	87.000000	14.160000	87.000000	
50%	100.000000	17.120000	100.000000	
75%	113.000000	20.000000	114.000000	
max	170.000000	30.910000	175.000000	

	total_night_charge	total_intl_calls	total_intl_charge	\
count	3294.000000	3294.000000	3294.000000	
mean	9.039997	4.449909	2.766381	
std	2.279285	2.379896	0.753333	
min	1.040000	0.000000	0.000000	
25%	7.520000	3.000000	2.300000	
50%	9.050000	4.000000	2.780000	
75%	10.590000	6.000000	3.270000	
max	17.770000	15.000000	5.400000	

	number_customer_service_calls	Target	area_code_408	\
count	3294.000000	3294.000000	3294.000000	
mean	1.538251	0.142684	0.251366	
std	1.263215	0.349803	0.433864	
min	0.000000	0.000000	0.000000	
25%	1.000000	0.000000	0.000000	
50%	1.000000	0.000000	0.000000	

```
75%                          2.000000      0.000000        1.000000
max                          6.000000      1.000000        1.000000

       area_code_415  area_code_510
count    3294.000000    3294.000000
mean        0.496357       0.252277
std         0.500063       0.434385
min         0.000000       0.000000
25%         0.000000       0.000000
50%         0.000000       0.000000
75%         1.000000       1.000000
max         1.000000       1.000000
```

[85]:
```
# Split dataset - Output variable
y = dados_treino.Target
```

[86]:
```
y
```

[86]:
```
0       0
1       0
2       0
3       0
4       0
       ..
3328    0
3329    0
3330    0
3331    0
3332    0
Name: Target, Length: 3294, dtype: int64
```

[87]:
```
# Creates a separate object for the input variables
X = dados_treino.drop('Target', axis = 1)
```

[88]:
```
X
```

[88]:
```
      account_length  international_plan  voice_mail_plan  \
0                128                   0                1
1                107                   0                1
2                137                   0                0
3                 84                   1                0
4                 75                   1                0
…                 …                    …                …
3328             192                   0                1
3329              68                   0                0
3330              28                   0                0
3331             184                   1                0
```

```
3332              74                    0                1

      number_vmail_messages  total_day_calls  total_day_charge  \
0                        25              110             45.07
1                        26              123             27.47
2                         0              114             41.38
3                         0               71             50.90
4                         0              113             28.34
…                       …                …                …
3328                     36               77             26.55
3329                      0               57             39.29
3330                      0              109             30.74
3331                      0              105             36.35
3332                     25              113             39.85

      total_eve_calls  total_eve_charge  total_night_calls  \
0                  99             16.78                 91
1                 103             16.62                103
2                 110             10.30                104
3                  88              5.26                 89
4                 122             12.61                121
…                 …                …                    …
3328              126             18.32                 83
3329               55             13.04                123
3330               58             24.55                 91
3331               84             13.57                137
3332               82             22.60                 77

      total_night_charge  total_intl_calls  total_intl_charge  \
0                  11.01                 3               2.70
1                  11.45                 3               3.70
2                   7.32                 5               3.29
3                   8.86                 7               1.78
4                   8.41                 3               2.73
…                   …                  …                 …
3328               12.56                 6               2.67
3329                8.61                 4               2.59
3330                8.64                 6               3.81
3331                6.26                10               1.35
3332               10.86                 4               3.70

      number_customer_service_calls  area_code_408  area_code_415  \
0                                 1              0              1
1                                 1              0              1
2                                 0              0              1
3                                 2              1              0
4                                 3              0              1
```

```
…                                …              …              …
3328                             2              0              1
3329                             3              0              1
3330                             2              0              0
3331                             2              0              0
3332                             0              0              1

      area_code_510
0                 0
1                 0
2                 0
3                 0
4                 0
…               …
3328              0
3329              0
3330              1
3331              1
3332              0

[3294 rows x 16 columns]
```

[89]:
```
print(X.shape, y.shape)
```

```
(3294, 16) (3294,)
```

[90]:
```
X_treino = X
```

[91]:
```
y_treino = y
```

1.7 Balancing the classes in the training dataset

[92]:
```
y.value_counts()
```

[92]:
```
0    2824
1     470
Name: Target, dtype: int64
```

[93]:
```
# Installs the package
!pip install -q imblearn
```

[94]:
```
# Load the SMOTE function
import imblearn
from imblearn.over_sampling import SMOTE
```

[95]:
```
# Let's apply the oversampling technique and increase the number of examples of␣
 ↪the minority class
over_sampler = SMOTE(k_neighbors = 2)
```

[96]:
```
# Apply oversampling (must be done with training data only)
X_res, y_res = over_sampler.fit_resample(X_treino, y_treino)
```

[97]:
```
y_res.value_counts()
```

[97]:
```
0    2824
1    2824
Name: Target, dtype: int64
```

[98]:
```
X_treino = X_res
y_treino = y_res
```

1.8 Standardization of the training dataset

[99]:
```
X_treino.head()
```

[99]:

	account_length	international_plan	voice_mail_plan	number_vmail_messages	total_day_calls	total_day_charge	total_eve_calls	total_eve_charge	total_night_calls	total_night_charge	total_intl_calls	total_intl_charge	number_customer_service_calls	area_code_408	area_code_415	area_code_510
0	128	0	1	25	110	45.07	99	16.78	91	11.01	3	2.70	1	0	1	0
1	107	0	1	26	123	27.47	103	16.62	103	11.45	3	3.70	1	0	1	0
2	137	0	0	0	114	41.38	110	10.30	104	7.32	5	3.29	0	0	1	0
3	84	1	0	0	71	50.90	88	5.26	89	8.86	7	1.78	2	1	0	0
4	75	1	0	0	113	28.34	122	12.61	121	8.41	3	2.73	3	0	1	0

[100]:
```
# We calculate mean and standard deviation of the training data
treino_mean = X_treino.mean()
treino_std = X_treino.std()
```

```
print(treino_mean)
print(treino_std)
```

```
account_length                  101.010623
international_plan                0.097911
voice_mail_plan                   0.218130
number_vmail_messages             6.659703
total_day_calls                 100.915368
total_day_charge                 32.662476
total_eve_calls                  99.657401
total_eve_charge                 17.550079
total_night_calls                99.966891
total_night_charge                9.126169
total_intl_calls                  4.124823
total_intl_charge                 2.827100
number_customer_service_calls     1.606941
area_code_408                     0.168024
area_code_415                     0.384030
area_code_510                     0.181480
dtype: float64
account_length                  39.269637
international_plan               0.297220
voice_mail_plan                  0.413013
number_vmail_messages           12.806414
total_day_calls                 19.720239
total_day_charge                10.169787
total_eve_calls                 19.137716
total_eve_charge                 4.092387
total_night_calls               19.142540
total_night_charge               2.091369
total_intl_calls                 2.241605
total_intl_charge                0.701778
number_customer_service_calls    1.332955
area_code_408                    0.373921
area_code_415                    0.486408
area_code_510                    0.385450
dtype: float64
```

[101]:
```
# Standardization
X_treino = (X_treino - treino_mean) / treino_std
```

[102]:
```
X_treino.head()
```

[102]:

	account_length	international_plan	voice_mail_plan	number_vmail_messages	\
0	0.687284	-0.329422	1.893088	1.432118	
1	0.152519	-0.329422	1.893088	1.510204	
2	0.916468	-0.329422	-0.528144	-0.520029	
3	-0.433175	3.035087	-0.528144	-0.520029	

```
4       -0.662360            3.035087       -0.528144             -0.520029

   total_day_calls  total_day_charge  total_eve_calls  total_eve_charge  \
0         0.460676          1.220038        -0.034351         -0.188174
1         1.119897         -0.510579         0.174660         -0.227271
2         0.663513          0.857198         0.540430         -1.771602
3        -1.516988          1.793304        -0.609132         -3.003157
4         0.612804         -0.425031         1.167464         -1.207139

   total_night_calls  total_night_charge  total_intl_calls  total_intl_charge  \
0          -0.468427            0.900765         -0.501794          -0.181112
1           0.158449            1.111153         -0.501794           1.243840
2           0.210688           -0.863630          0.390424           0.659610
3          -0.572907           -0.127270          1.282642          -1.492068
4           1.098763           -0.342440         -0.501794          -0.138363

   number_customer_service_calls  area_code_408  area_code_415  area_code_510
0                      -0.455335      -0.449357       1.266365      -0.470827
1                      -0.455335      -0.449357       1.266365      -0.470827
2                      -1.205548      -0.449357       1.266365      -0.470827
3                       0.294878       2.225006      -0.789522      -0.470827
4                       1.045092      -0.449357       1.266365      -0.470827
```

[103]:
```
# Describe
X_treino.describe()
```

[103]:
```
       account_length  international_plan  voice_mail_plan  \
count    5.648000e+03        5.648000e+03     5.648000e+03
mean    -3.917625e-17       -1.111599e-16    -9.375350e-15
std      1.000000e+00        1.000000e+00     1.000000e+00
min     -2.546767e+00       -3.294216e-01    -5.281440e-01
25%     -6.878246e-01       -3.294216e-01    -5.281440e-01
50%     -2.705202e-04       -3.294216e-01    -5.281440e-01
75%      6.618186e-01       -3.294216e-01    -5.281440e-01
max      2.953666e+00        3.035087e+00     1.893088e+00

       number_vmail_messages  total_day_calls  total_day_charge  \
count           5.648000e+03     5.648000e+03      5.648000e+03
mean           -3.971151e-15    -1.502968e-16      1.867044e-15
std             1.000000e+00     1.000000e+00      1.000000e+00
min            -5.200287e-01    -3.088977e+00     -3.168451e+00
25%            -5.200287e-01    -6.549296e-01     -7.347721e-01
50%            -5.200287e-01     4.291618e-03     -2.580936e-02
75%            -5.200287e-01     7.142222e-01      7.865557e-01
max             3.462351e+00     2.844014e+00      2.652713e+00

       total_eve_calls  total_eve_charge  total_night_calls  \
```

```
count     5.648000e+03      5.648000e+03       5.648000e+03
mean     -2.785681e-16     -1.013989e-14       2.341336e-16
std       1.000000e+00      1.000000e+00       1.000000e+00
min      -5.207382e+00     -4.288470e+00      -3.498328e+00
25%      -6.613851e-01     -6.817731e-01      -7.296258e-01
50%       1.790178e-02      2.025127e-02       1.729607e-03
75%       6.971887e-01      6.988392e-01       6.808453e-01
max       3.675600e+00      3.264579e+00       3.919705e+00

       total_night_charge  total_intl_calls  total_intl_charge  \
count        5.648000e+03      5.648000e+03       5.648000e+03
mean        -3.226094e-16      3.205081e-15       3.800759e-15
std          1.000000e+00      1.000000e+00       1.000000e+00
min         -3.866448e+00     -1.840121e+00      -4.028482e+00
25%         -6.723678e-01     -5.017936e-01      -6.212106e-01
50%          4.794559e-03     -5.568464e-02       1.838125e-02
75%          6.852684e-01      3.904243e-01       6.596095e-01
max          4.133098e+00      4.851514e+00       3.666258e+00

       number_customer_service_calls  area_code_408  area_code_415  \
count                   5.648000e+03   5.648000e+03   5.648000e+03
mean                    6.281960e-16  -1.813882e-15  -6.223500e-15
std                     1.000000e+00   1.000000e+00   1.000000e+00
min                    -1.205548e+00  -4.493574e-01  -7.895218e-01
25%                    -4.553347e-01  -4.493574e-01  -7.895218e-01
50%                    -4.553347e-01  -4.493574e-01  -7.895218e-01
75%                     2.948784e-01  -4.493574e-01   1.266365e+00
max                     3.295731e+00   2.225006e+00   1.266365e+00

       area_code_510
count   5.648000e+03
mean   -6.420305e-15
std     1.000000e+00
min    -4.708272e-01
25%    -4.708272e-01
50%    -4.708272e-01
75%    -4.708272e-01
max     2.123545e+00
```

2 Preparing the test data

[104]:
```
dados_treino.columns
```

[104]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
```

```
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target', 'area_code_408',
       'area_code_415', 'area_code_510'],
      dtype='object')
```

[105]:
```
dados_teste.shape
```

[105]:
```
(1667, 21)
```

[106]:
```
#Rename the variable target
dados_teste.rename({'churn':'Target'}, axis = 'columns', inplace = True)
```

[107]:
```
dados_teste.columns
```

[107]:
```
Index(['Unnamed: 0', 'state', 'account_length', 'area_code',
       'international_plan', 'voice_mail_plan', 'number_vmail_messages',
       'total_day_minutes', 'total_day_calls', 'total_day_charge',
       'total_eve_minutes', 'total_eve_calls', 'total_eve_charge',
       'total_night_minutes', 'total_night_calls', 'total_night_charge',
       'total_intl_minutes', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target'],
      dtype='object')
```

[108]:
```
# Applies the function
dados_teste['Target'] = dados_teste['Target'].map(encoding_func)
```

[109]:
```
dados_teste.sample(5)
```

[109]:

	Unnamed: 0	state	account_length	area_code	international_plan	\
486	487	WY	117	area_code_415	no	
997	998	KY	76	area_code_510	no	
221	222	WI	72	area_code_415	no	
853	854	WV	63	area_code_415	no	
1490	1491	RI	77	area_code_408	no	

	voice_mail_plan	number_vmail_messages	total_day_minutes	\
486	no	0	234.8	
997	no	0	293.8	
221	no	0	129.8	
853	no	0	147.8	
1490	no	0	196.6	

	total_day_calls	total_day_charge	…	total_eve_calls	\
486	118	39.92	…	112	
997	94	49.95	…	62	
221	106	22.07	…	138	
853	95	25.13	…	72	
1490	87	33.42	…	103	

```
      total_eve_charge  total_night_minutes  total_night_calls  \
486              17.99                147.0                 97
997              14.42                160.4                103
221              16.04                212.5                116
853              17.12                154.9                118
1490             10.00                124.9                100

      total_night_charge  total_intl_minutes  total_intl_calls  \
486                 6.61                 7.5                 3
997                 7.22                13.7                 9
221                 9.56                 8.3                 4
853                 6.97                 8.6                 8
1490                5.62                11.4                 3

      total_intl_charge  number_customer_service_calls  Target
486                2.03                              2       0
997                3.70                              0       1
221                2.24                              4       1
853                2.32                              1       0
1490               3.08                              2       0

[5 rows x 21 columns]
```

[110]:
```
# Function for label encoding for international_plan -> 0 = no and 1 = yes
# Apply function
dados_teste['international_plan'] = dados_teste['international_plan'].
 ↪map(encoding_func)
```

[111]:
```
# Function for label encoding for voice_mail_plan -> 0 = no and 1 = yes
# Apply the function
dados_teste['voice_mail_plan'] = dados_teste['voice_mail_plan'].
 ↪map(encoding_func)
```

[112]:
```
dados_teste.sample(5)
```

[112]:
```
      Unnamed: 0 state  account_length      area_code  international_plan  \
586          587    ID              47  area_code_415                   0
424          425    NH              44  area_code_510                   0
1401        1402    PA              86  area_code_415                   0
1452        1453    ID             127  area_code_408                   0
308          309    KY             122  area_code_415                   0

      voice_mail_plan  number_vmail_messages  total_day_minutes  \
586                 1                     28              196.2
424                 1                     25              152.9
1401                1                     23              247.6
```

```
1452                0                      0              189.7
308                 0                      0              128.9

      total_day_calls  total_day_charge  …  total_eve_calls  \
586                88             33.35  …              106
424               106             25.99  …              147
1401               65             42.09  …              104
1452              110             32.25  …              116
308               136             21.91  …              133

      total_eve_charge  total_night_minutes  total_night_calls  \
586              16.56                243.0                103
424              17.47                247.3                107
1401             21.73                150.0                138
1452             12.10                175.2                 79
308              20.77                148.7                133

      total_night_charge  total_intl_minutes  total_intl_calls  \
586                10.93                10.5                10
424                11.13                 8.0                 5
1401                6.75                 7.3                 5
1452                7.88                 8.5                 8
308                 6.69                10.7                 5

      total_intl_charge  number_customer_service_calls  Target
586                2.84                              0       0
424                2.16                              1       0
1401               1.97                              3       0
1452               2.30                              2       0
308                2.89                              0       0

[5 rows x 21 columns]
```

[113]:
```
# Applying One-Hot Encoding
for cat in ['area_code']:
    onehots = pd.get_dummies(dados_teste[cat], prefix = cat)
    dados_teste = dados_teste.join(onehots)
```

[114]:
```
dados_teste.columns
```

[114]:
```
Index(['Unnamed: 0', 'state', 'account_length', 'area_code',
       'international_plan', 'voice_mail_plan', 'number_vmail_messages',
       'total_day_minutes', 'total_day_calls', 'total_day_charge',
       'total_eve_minutes', 'total_eve_calls', 'total_eve_charge',
       'total_night_minutes', 'total_night_calls', 'total_night_charge',
       'total_intl_minutes', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target', 'area_code_area_code_408',
```

```
       'area_code_area_code_415', 'area_code_area_code_510'],
      dtype='object')
```

[115]:
```
dados_teste = dados_teste.drop(columns = ['Unnamed: 0',
                          'state',
                          'area_code'])
```

[116]:
```
# Removing the total_day_minutes, total_eve_minutes, total_night_minutes and␣
 ↪total_intl_minutes columns to avoid correlation
dados_teste = dados_teste.drop(columns = ['total_day_minutes',
                                          'total_eve_minutes',
                                          'total_night_minutes',
                                          'total_intl_minutes'])
```

[117]:
```
#Rename the variable area_code
dados_teste.rename({'area_code_area_code_408':
 ↪'area_code_408','area_code_area_code_415':
 ↪'area_code_415','area_code_area_code_510':'area_code_510'}, axis =␣
 ↪'columns', inplace = True)
```

[118]:
```
dados_teste.columns
```

[118]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target', 'area_code_408',
       'area_code_415', 'area_code_510'],
      dtype='object')
```

[119]:
```
y_teste = dados_teste.Target
```

[120]:
```
# Creates a separate object for the input variables
X_teste = dados_teste.drop('Target', axis = 1)
```

[121]:
```
# We use training mean and variance to standardize the test data set
X_teste = (X_teste - treino_mean) / treino_std
```

3 Logistic Regression Model

[122]:
```
# Set hyperparameter list
tuned_params_v1 = {'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000],
                   'penalty': ['l1', 'l2']}
```

[123]:
```
# We will create the model with GridSearch
# Several models will be created with different combinations of hyperparameters
modelo_v1 = GridSearchCV(LogisticRegression(),
```

```
                          tuned_params_v1,
                          scoring = 'roc_auc',
                          n_jobs = -1)
```

[124]:
```
# Model training
modelo_v1.fit(X_treino, y_treino)
```

[124]:
```
GridSearchCV(estimator=LogisticRegression(), n_jobs=-1,
             param_grid={'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000,
                               10000],
                         'penalty': ['l1', 'l2']},
             scoring='roc_auc')
```

[125]:
```
# We select the best model
modelo_v1.best_estimator_
```

[125]:
```
LogisticRegression(C=1)
```

[126]:
```
# Predictions with test data
y_pred_v1 = modelo_v1.predict(X_teste)
```

[127]:
```
# Show the top 10 predictions
y_pred_v1[:10]
```

[127]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

[128]:
```
# We get the predictions in probability format for each class
y_pred_proba_v1 = modelo_v1.predict_proba(X_teste)
```

[129]:
```
# Show the top 10 predictions
y_pred_proba_v1[:10]
```

[129]:
```
array([[0.76782906, 0.23217094],
       [0.84187671, 0.15812329],
       [0.64092716, 0.35907284],
       [0.50128711, 0.49871289],
       [0.76635499, 0.23364501],
       [0.76183735, 0.23816265],
       [0.70511519, 0.29488481],
       [0.86494836, 0.13505164],
       [0.90046646, 0.09953354],
       [0.78185034, 0.21814966]])
```

[130]:
```
# We get the predictions in probability format by filtering for the positive␣
 ↪class
# We need this to calculate the ROC curve
y_pred_proba_v1 = modelo_v1.predict_proba(X_teste)[:,1]
```

[131]:
```
# Show the top 10 predictions
y_pred_proba_v1[:10]
```

[131]:
```
array([0.23217094, 0.15812329, 0.35907284, 0.49871289, 0.23364501,
       0.23816265, 0.29488481, 0.13505164, 0.09953354, 0.21814966])
```

[132]:
```
# As an example, let's check one of the data points (change the value of i if␣
 ↪you wish)
i = 16
print('For data point {}, actual class = {}, predicted class = {}, predicted␣
 ↪probability = {}'.
      format(i, y_teste.iloc[i], y_pred_v1[i], y_pred_proba_v1[i]))
```

```
For data point 16, actual class = 0, predicted class = 0, predicted probability
= 0.06632062821105963
```

[133]:
```
# Confusion matrix
confusion_matrix(y_teste, y_pred_v1)
```

[133]:
```
array([[1276,  167],
       [ 109,  115]], dtype=int64)
```

[134]:
```
# Extracting each value from the CM
tn, fp, fn, tp = confusion_matrix(y_teste, y_pred_v1).ravel()
```

[135]:
```
print(tn, fp, fn, tp)
```

```
1276 167 109 115
```

[136]:
```
# Calculate overall AUC (Area Under The Curve) metric with actual data and␣
 ↪predictions under test
roc_auc_v1 = roc_auc_score(y_teste, y_pred_v1)
print(roc_auc_v1)
```

```
0.6988308707058708
```

[137]:
```
# Calculate the ROC curve with data and predictions under test
fpr_v1, tpr_v1, thresholds = roc_curve(y_teste, y_pred_proba_v1)
```

[138]:
```
# AUC in test
auc_v1 = auc(fpr_v1, tpr_v1)
print(auc_v1)
```

```
0.8137344074844074
```

[139]:
```
# Test Accuracy
acuracia_v1 = accuracy_score(y_teste, y_pred_v1)
print(acuracia_v1)
```

```
0.8344331133773245
```

3.0.1 Feature Importance

[140]:
```
# Building the model again with the best hyperparameters
# This is necessary because the final version should not have GridSearchCV
modelo_v1 = LogisticRegression(C = 1)
modelo_v1.fit(X_treino, y_treino)
```

[140]:
```
LogisticRegression(C=1)
```

[141]:
```
# We get the coefficients by largest using np.argsort
indices = np.argsort(-abs(modelo_v1.coef_[0,:]))
```

[142]:
```
print("Most important variables for the model result_v1:")
print(50*'-')
for feature in X.columns[indices]:
    print(feature)
```

```
Most important variables for the model result_v1:
--------------------------------------------------
area_code_415
area_code_408
area_code_510
voice_mail_plan
number_vmail_messages
total_day_charge
number_customer_service_calls
international_plan
total_eve_charge
total_intl_calls
total_intl_charge
total_night_charge
total_eve_calls
account_length
total_night_calls
total_day_calls
```

[144]:
```
# Save the template to disk
with open('modelos/modelo_regressao.pkl', 'wb') as pickle_file:
      joblib.dump(modelo_v1, 'modelos/modelo_regressao.pkl')
```

3.1 Model V2

[145]:
```
dados_treino.columns
```

[145]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
```

```
       'number_customer_service_calls', 'Target', 'area_code_408',
       'area_code_415', 'area_code_510'],
      dtype='object')
```

[146]:
```
# Removing the total_day_minutes, total_eve_minutes, total_night_minutes and
 ↪total_intl_minutes columns to avoid correlation
dados_treino = dados_teste.drop(columns = ['area_code_408',
                                           'area_code_510',
                                           'area_code_415'])
```

[147]:
```
dados_treino.columns
```

[147]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'Target'],
      dtype='object')
```

[148]:
```
dados_treino.sample(5)
```

[148]:

	account_length	international_plan	voice_mail_plan	\
1605	80	0	0	
692	127	0	1	
1064	69	0	0	
280	78	0	0	
74	93	0	0	

	number_vmail_messages	total_day_calls	total_day_charge	\
1605	0	109	35.68	
692	24	121	35.96	
1064	0	88	37.67	
280	0	119	49.39	
74	0	104	20.50	

	total_eve_calls	total_eve_charge	total_night_calls	\
1605	56	19.22	121	
692	115	15.63	103	
1064	87	19.70	116	
280	75	13.46	101	
74	95	17.47	107	

	total_night_charge	total_intl_calls	total_intl_charge	\
1605	6.94	5	2.43	
692	8.38	1	3.65	
1064	9.53	9	2.32	
280	11.50	6	2.32	

```
74                  8.21                  6               2.59

      number_customer_service_calls  Target
1605                              1       0
692                               3       0
1064                              1       0
280                               2       1
74                                2       0
```

[149]:
```
# Describe
X_treino.describe()
```

[149]:
```
       account_length  international_plan  voice_mail_plan  \
count    5.648000e+03        5.648000e+03     5.648000e+03
mean    -3.917625e-17       -1.111599e-16    -9.375350e-15
std      1.000000e+00        1.000000e+00     1.000000e+00
min     -2.546767e+00       -3.294216e-01    -5.281440e-01
25%     -6.878246e-01       -3.294216e-01    -5.281440e-01
50%     -2.705202e-04       -3.294216e-01    -5.281440e-01
75%      6.618186e-01       -3.294216e-01    -5.281440e-01
max      2.953666e+00        3.035087e+00     1.893088e+00

       number_vmail_messages  total_day_calls  total_day_charge  \
count           5.648000e+03     5.648000e+03      5.648000e+03
mean           -3.971151e-15    -1.502968e-16      1.867044e-15
std             1.000000e+00     1.000000e+00      1.000000e+00
min            -5.200287e-01    -3.088977e+00     -3.168451e+00
25%            -5.200287e-01    -6.549296e-01     -7.347721e-01
50%            -5.200287e-01     4.291618e-03     -2.580936e-02
75%            -5.200287e-01     7.142222e-01      7.865557e-01
max             3.462351e+00     2.844014e+00      2.652713e+00

       total_eve_calls  total_eve_charge  total_night_calls  \
count     5.648000e+03      5.648000e+03       5.648000e+03
mean     -2.785681e-16     -1.013989e-14       2.341336e-16
std       1.000000e+00      1.000000e+00       1.000000e+00
min      -5.207382e+00     -4.288470e+00      -3.498328e+00
25%      -6.613851e-01     -6.817731e-01      -7.296258e-01
50%       1.790178e-02      2.025127e-02       1.729607e-03
75%       6.971887e-01      6.988392e-01       6.808453e-01
max       3.675600e+00      3.264579e+00       3.919705e+00

       total_night_charge  total_intl_calls  total_intl_charge  \
count        5.648000e+03      5.648000e+03       5.648000e+03
mean        -3.226094e-16      3.205081e-15       3.800759e-15
std          1.000000e+00      1.000000e+00       1.000000e+00
min         -3.866448e+00     -1.840121e+00      -4.028482e+00
```

```
25%          -6.723678e-01     -5.017936e-01      -6.212106e-01
50%           4.794559e-03     -5.568464e-02       1.838125e-02
75%           6.852684e-01      3.904243e-01       6.596095e-01
max           4.133098e+00      4.851514e+00       3.666258e+00

       number_customer_service_calls  area_code_408  area_code_415  \
count                   5.648000e+03   5.648000e+03   5.648000e+03
mean                    6.281960e-16  -1.813882e-15  -6.223500e-15
std                     1.000000e+00   1.000000e+00   1.000000e+00
min                    -1.205548e+00  -4.493574e-01  -7.895218e-01
25%                    -4.553347e-01  -4.493574e-01  -7.895218e-01
50%                    -4.553347e-01  -4.493574e-01  -7.895218e-01
75%                     2.948784e-01  -4.493574e-01   1.266365e+00
max                     3.295731e+00   2.225006e+00   1.266365e+00

       area_code_510
count   5.648000e+03
mean   -6.420305e-15
std     1.000000e+00
min    -4.708272e-01
25%    -4.708272e-01
50%    -4.708272e-01
75%    -4.708272e-01
max     2.123545e+00
```

[150]:
```
# Describe
y_treino.describe()
```

[150]:
```
count    5648.000000
mean        0.500000
std         0.500044
min         0.000000
25%         0.000000
50%         0.500000
75%         1.000000
max         1.000000
Name: Target, dtype: float64
```

[151]:
```
# Removing the total_day_minutes, total_eve_minutes, total_night_minutes and␣
 ↪total_intl_minutes columns to avoid correlation
X_treino = X_treino.drop(columns = ['area_code_408',
                                    'area_code_510',
                                    'area_code_415'])
```

[152]:
```
# Describe
X_treino.describe()
```

[152]:

	account_length	international_plan	voice_mail_plan
count	5.648000e+03	5.648000e+03	5.648000e+03
mean	-3.917625e-17	-1.111599e-16	-9.375350e-15
std	1.000000e+00	1.000000e+00	1.000000e+00
min	-2.546767e+00	-3.294216e-01	-5.281440e-01
25%	-6.878246e-01	-3.294216e-01	-5.281440e-01
50%	-2.705202e-04	-3.294216e-01	-5.281440e-01
75%	6.618186e-01	-3.294216e-01	-5.281440e-01
max	2.953666e+00	3.035087e+00	1.893088e+00

	number_vmail_messages	total_day_calls	total_day_charge
count	5.648000e+03	5.648000e+03	5.648000e+03
mean	-3.971151e-15	-1.502968e-16	1.867044e-15
std	1.000000e+00	1.000000e+00	1.000000e+00
min	-5.200287e-01	-3.088977e+00	-3.168451e+00
25%	-5.200287e-01	-6.549296e-01	-7.347721e-01
50%	-5.200287e-01	4.291618e-03	-2.580936e-02
75%	-5.200287e-01	7.142222e-01	7.865557e-01
max	3.462351e+00	2.844014e+00	2.652713e+00

	total_eve_calls	total_eve_charge	total_night_calls
count	5.648000e+03	5.648000e+03	5.648000e+03
mean	-2.785681e-16	-1.013989e-14	2.341336e-16
std	1.000000e+00	1.000000e+00	1.000000e+00
min	-5.207382e+00	-4.288470e+00	-3.498328e+00
25%	-6.613851e-01	-6.817731e-01	-7.296258e-01
50%	1.790178e-02	2.025127e-02	1.729607e-03
75%	6.971887e-01	6.988392e-01	6.808453e-01
max	3.675600e+00	3.264579e+00	3.919705e+00

	total_night_charge	total_intl_calls	total_intl_charge
count	5.648000e+03	5.648000e+03	5.648000e+03
mean	-3.226094e-16	3.205081e-15	3.800759e-15
std	1.000000e+00	1.000000e+00	1.000000e+00
min	-3.866448e+00	-1.840121e+00	-4.028482e+00
25%	-6.723678e-01	-5.017936e-01	-6.212106e-01
50%	4.794559e-03	-5.568464e-02	1.838125e-02
75%	6.852684e-01	3.904243e-01	6.596095e-01
max	4.133098e+00	4.851514e+00	3.666258e+00

	number_customer_service_calls
count	5.648000e+03
mean	6.281960e-16
std	1.000000e+00
min	-1.205548e+00
25%	-4.553347e-01
50%	-4.553347e-01

```
75%                    2.948784e-01
max                    3.295731e+00
```

3.1.1 Do the same thing with Test Data

[153]:
```
X_teste.sample(5)
```

[153]:
```
      account_length  international_plan  voice_mail_plan  \
1027        1.451233           -0.329422        -0.528144
331        -0.076665           -0.329422        -0.528144
1263       -0.229455           -0.329422         1.893088
116         0.865538           -0.329422         1.893088
1255       -0.509570           -0.329422         1.893088

      number_vmail_messages  total_day_calls  total_day_charge  \
1027              -0.520029         1.119897         -1.208725
331               -0.520029         1.373443         -1.668912
1263               1.119775         0.308548          0.596623
116                0.885517        -0.553511         -0.200838
1255               2.837664         0.055001         -1.499783

      total_eve_calls  total_eve_charge  total_night_calls  \
1027         1.167464          1.158229           0.576366
331         -0.556879          0.906542           0.524126
1263        -1.549683         -1.040977          -1.460981
116         -1.810948         -1.121614          -0.781865
1255         0.226913          0.390950          -0.154989

      total_night_charge  total_intl_calls  total_intl_charge  \
1027            0.365230          0.390424           0.745107
331             0.197876          5.297623           0.132377
1263            1.225911         -0.501794           0.745107
116             1.192440          2.174860           0.859103
1255           -0.442853          0.390424          -1.292574

      number_customer_service_calls  area_code_408  area_code_415  \
1027                      -0.455335       2.225006      -0.789522
331                        1.045092       2.225006      -0.789522
1263                      -1.205548       2.225006      -0.789522
116                       -0.455335      -0.449357      -0.789522
1255                       0.294878      -0.449357       1.266365

      area_code_510
1027      -0.470827
331       -0.470827
1263      -0.470827
116        2.123545
```

```
1255      -0.470827
```

[154]:
```
# Removing the total_day_minutes, total_eve_minutes, total_night_minutes and␣
↪total_intl_minutes columns to avoid correlation
X_teste = X_teste.drop(columns = ['area_code_408',
                                  'area_code_510',
                                  'area_code_415'])
```

3.2 Prediction V2

[155]:
```
# Define hyperparameter list
tuned_params_v1 = {'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000],
                   'penalty': ['l1', 'l2']}
```

[156]:
```
# We will create the model with GridSearch
# Several models will be created with different combinations of hyperparameters
modelo_v1 = GridSearchCV(LogisticRegression(),
                         tuned_params_v1,
                         scoring = 'roc_auc',
                         n_jobs = -1)
```

[157]:
```
# Model training
modelo_v1.fit(X_treino, y_treino)
```

[157]:
```
GridSearchCV(estimator=LogisticRegression(), n_jobs=-1,
             param_grid={'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000,
                               10000],
                         'penalty': ['l1', 'l2']},
             scoring='roc_auc')
```

[158]:
```
# We select the best model
modelo_v1.best_estimator_
```

[158]:
```
LogisticRegression(C=1000)
```

[159]:
```
# Show the top 10 predictions
y_pred_v1[:10]
```

[159]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

[160]:
```
# We get the predictions in probability format for each class
y_pred_proba_v1 = modelo_v1.predict_proba(X_teste)
```

[161]:
```
# Show the top 10 predictions
y_pred_proba_v1[:10]
```

[161]:
```
array([[0.60666656, 0.39333344],
       [0.71515418, 0.28484582],
```

```
       [0.35897875, 0.64102125],
       [0.32626946, 0.67373054],
       [0.65828174, 0.34171826],
       [0.6687996 , 0.3312004 ],
       [0.48976139, 0.51023861],
       [0.71391601, 0.28608399],
       [0.81799053, 0.18200947],
       [0.5599571 , 0.4400429 ]])
```

[162]:
```
# We get the predictions in probability format by filtering for the positive class
# We need this to calculate the ROC curve
y_pred_proba_v1 = modelo_v1.predict_proba(X_teste)[:,1]
```

[163]:
```
# Show the top 10 predictions
y_pred_proba_v1[:10]
```

[163]:
```
array([0.39333344, 0.28484582, 0.64102125, 0.67373054, 0.34171826,
       0.3312004 , 0.51023861, 0.28608399, 0.18200947, 0.4400429 ])
```

[164]:
```
# As an example, let's check one of the data points (change the value of i if you wish)
i = 16
print('For data point {}, actual class = {}, predicted class = {}, predicted probability = {}'.
      format(i, y_teste.iloc[i], y_pred_v1[i], y_pred_proba_v1[i]))
```

```
For data point 16, actual class = 0, predicted class = 0, predicted probability
= 0.1300442517053336
```

[165]:
```
# Confusion matrix
confusion_matrix(y_teste, y_pred_v1)
```

[165]:
```
array([[1276,  167],
       [ 109,  115]], dtype=int64)
```

[166]:
```
# Extracting each value from the CM
tn, fp, fn, tp = confusion_matrix(y_teste, y_pred_v1).ravel()
```

[167]:
```
print(tn, fp, fn, tp)
```

```
1276 167 109 115
```

[168]:
```
# Calculate overall AUC (Area Under The Curve) metric with actual data and predictions under test
roc_auc_v1 = roc_auc_score(y_teste, y_pred_v1)
print(roc_auc_v1)
```

```
0.6988308707058708
```

[169]:
```
# Calculate overall AUC (Area Under The Curve) metric with actual data and␣
 ↪predictions under test
fpr_v1, tpr_v1, thresholds = roc_curve(y_teste, y_pred_proba_v1)
```

[170]:
```
# AUC in test
auc_v1 = auc(fpr_v1, tpr_v1)
print(auc_v1)
```

```
0.8082213394713396
```

[171]:
```
# Test Accuracy
acuracia_v1 = accuracy_score(y_teste, y_pred_v1)
print(acuracia_v1)
```

```
0.8344331133773245
```

3.3 Model V1 with 5 variables

[172]:
```
'''international_plan, voice_mail_plan, total_day_charge, total_eve_charge,␣
 ↪number_customer

voice_mail_plan
total_day_charge
number_customer_service_calls
international_plan
number_vmail_messages'''
```

[172]:
```
'international_plan, voice_mail_plan, total_day_charge, total_eve_charge, number
_customer\n\nvoice_mail_plan\ntotal_day_charge\nnumber_customer_service_calls\ni
nternational_plan\nnumber_vmail_messages'
```

[173]:
```
X_treino.columns
```

[173]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls'],
      dtype='object')
```

[174]:
```
X_treino = X_treino.drop(columns = ['account_length',
                                    'total_day_calls',
                                    'total_eve_calls',
                                    'total_eve_charge',
                                    'total_night_calls',
                                    'total_night_charge',
                                    'total_intl_calls',
                                    'total_intl_charge'])
```

[175]:
```
X_treino.columns
```

[175]:
```
Index(['international_plan', 'voice_mail_plan', 'number_vmail_messages',
       'total_day_charge', 'number_customer_service_calls'],
      dtype='object')
```

[176]:
```
X_teste.columns
```

[176]:
```
Index(['account_length', 'international_plan', 'voice_mail_plan',
       'number_vmail_messages', 'total_day_calls', 'total_day_charge',
       'total_eve_calls', 'total_eve_charge', 'total_night_calls',
       'total_night_charge', 'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls'],
      dtype='object')
```

[177]:
```
X_teste = X_teste.drop(columns = ['account_length',
                                  'total_day_calls',
                                  'total_eve_calls',
                                  'total_eve_charge',
                                  'total_night_calls',
                                  'total_night_charge',
                                  'total_intl_calls',
                                  'total_intl_charge'])
```

[178]:
```
# Define hyperparameter list
tuned_params_v1 = {'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000],
                   'penalty': ['l2']}
```

[179]:
```
# We will create the model with GridSearch
# Several models will be created with different combinations of hyperparameters
modelo_v1 = GridSearchCV(LogisticRegression(),
                         tuned_params_v1,
                         scoring = 'roc_auc')
```

[180]:
```
# Model training
modelo_v1.fit(X_treino, y_treino)
```

[180]:
```
GridSearchCV(estimator=LogisticRegression(),
             param_grid={'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000,
                               10000],
                         'penalty': ['l2']},
             scoring='roc_auc')
```

[181]:
```
# We select the best model
modelo_v1.best_estimator_
```

[181]:
```
LogisticRegression(C=100)
```

[182]:
```
# Show the top 10 predictions
y_pred_v1[:10]
```

[182]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

[183]:
```
# We get the predictions in probability format for each class
y_pred_proba_v1 = modelo_v1.predict_proba(X_teste)
```

[184]:
```
# Show the top 10 predictions
y_pred_proba_v1[:10]
```

[184]:
```
array([[0.71619511, 0.28380489],
       [0.53060092, 0.46939908],
       [0.47326183, 0.52673817],
       [0.46221861, 0.53778139],
       [0.4328643 , 0.5671357 ],
       [0.4327771 , 0.5672229 ],
       [0.64520319, 0.35479681],
       [0.71773681, 0.28226319],
       [0.7424734 , 0.2575266 ],
       [0.65298512, 0.34701488]])
```

[185]:
```
# We get the predictions in probability format by filtering for the positive␣
 ↪class
# We need this to calculate the ROC curve
y_pred_proba_v1 = modelo_v1.predict_proba(X_teste)[:,1]
```

[186]:
```
# Show the top 10 predictions
y_pred_proba_v1[:10]
```

[186]:
```
array([0.28380489, 0.46939908, 0.52673817, 0.53778139, 0.5671357 ,
       0.5672229 , 0.35479681, 0.28226319, 0.2575266 , 0.34701488])
```

[187]:
```
# As an example, let's check one of the data points (change the value of i if␣
 ↪you wish)
i = 16
print('For data point {}, actual class = {}, predicted class = {}, predicted␣
 ↪probability = {}'.
      format(i, y_teste.iloc[i], y_pred_v1[i], y_pred_proba_v1[i]))
```

```
For data point 16, actual class = 0, predicted class = 0, predicted probability
= 0.0659158425735789
```

[188]:
```
# Confusion matrix
confusion_matrix(y_teste, y_pred_v1)
```

[188]:
```
array([[1276,  167],
       [ 109,  115]], dtype=int64)
```

[189]:
```
# Extracting each value from the CM
tn, fp, fn, tp = confusion_matrix(y_teste, y_pred_v1).ravel()
```

[190]:
```
# Calculate overall AUC (Area Under The Curve) metric with actual data and␣
↪predictions under test
roc_auc_v1 = roc_auc_score(y_teste, y_pred_v1)
print(roc_auc_v1)
```

```
0.6988308707058708
```

[191]:
```
# Calculate the ROC curve with data and predictions under test
fpr_v1, tpr_v1, thresholds = roc_curve(y_teste, y_pred_proba_v1)
```

[192]:
```
# AUC in test
auc_v1 = auc(fpr_v1, tpr_v1)
print(auc_v1)
```

```
0.8098919661419662
```

[193]:
```
# Test Accuracy
acuracia_v1 = accuracy_score(y_teste, y_pred_v1)
print(acuracia_v1)
```

```
0.8344331133773245
```

[194]:
```
# Create a dataframe to receive the metrics for each model
df_modelos = pd.DataFrame()
```

[195]:
```
# Dictionary with model_v1 metrics
dict_modelo_v1 = {'Nome': 'modelo_v1',
                  'Algoritmo': 'Regressão Logística',
                  'ROC_AUC Score': roc_auc_v1,
                  'AUC Score': auc_v1,
                  'Acurácia': acuracia_v1}
```

[196]:
```
# Add dict to dataframe
df_modelos = df_modelos.append(dict_modelo_v1, ignore_index = True)
```

[197]:
```
display(df_modelos)
```

	Nome	Algoritmo	ROC_AUC Score	AUC Score	Acurácia
0	modelo_v1	Regressão Logística	0.698831	0.809892	0.834433

4 Random Forest Model

[198]:
```
X_treino.sample(5)
```

[198]:

	international_plan	voice_mail_plan	number_vmail_messages	\
2304	-0.329422	-0.528144	-0.520029	

```
942            -0.329422         1.893088              2.525320
2254           -0.329422        -0.528144             -0.520029
4907           -0.329422         1.893088              1.197860
4191           -0.329422        -0.528144             -0.520029

      total_day_charge  number_customer_service_calls
2304         -0.790820                       0.294878
942          -0.274585                       0.294878
2254          0.584823                       2.545518
4907          0.115782                      -0.455335
4191         -1.495100                       2.545518
```

[199]:
```
# Hyperparameter grid
tuned_params_v2 = {'n_estimators': [100, 200, 300, 400, 500],
                   'min_samples_split': [2, 5, 10],
                   'min_samples_leaf': [1, 2, 4]}
```

[200]:
```
# Create the model with RandomizedSearchCV to search for the best combination␣
 ↪of hyperparameters
modelo_v2 = RandomizedSearchCV(RandomForestClassifier(),
                               tuned_params_v2,
                               n_iter = 15,
                               scoring = 'roc_auc',
                               n_jobs  = -1)
```

[201]:
```
# Model training
modelo_v2.fit(X_treino, y_treino)
```

[201]:
```
RandomizedSearchCV(estimator=RandomForestClassifier(), n_iter=15, n_jobs=-1,
                   param_distributions={'min_samples_leaf': [1, 2, 4],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [100, 200, 300, 400,
                                                         500]},
                   scoring='roc_auc')
```

[202]:
```
# Extract the best model
modelo_v2.best_estimator_
```

[202]:
```
RandomForestClassifier(min_samples_leaf=4, n_estimators=300)
```

[203]:
```
# Predictions under test
y_pred_v2 = modelo_v2.predict(X_teste)
```

[204]:
```
# Get the predictions for the positive class
y_pred_proba_v2 = modelo_v2.predict_proba(X_teste)[:,1]
```

[205]:
```
# Confusion matrix
confusion_matrix(y_teste, y_pred_v2)
```

[205]:
```
array([[1132,  311],
       [  56,  168]], dtype=int64)
```

[206]:
```
# ROC curve in data and predictions under test
roc_auc_v2 = roc_auc_score(y_teste, y_pred_v2)
print(roc_auc_v2)
```

```
0.7672383922383922
```

[207]:
```
# ROC curve in data and predictions under test
fpr_v2, tpr_v2, thresholds = roc_curve(y_teste, y_pred_proba_v2)
```

[208]:
```
# AUC in test
auc_v2 = auc(fpr_v2, tpr_v2)
print(auc_v2)
```

```
0.8406021062271062
```

[209]:
```
# Test Accuracy
acuracia_v2 = accuracy_score(y_teste, y_pred_v2)
print(acuracia_v2)
```

```
0.7798440311937612
```

[210]:
```
# Save the template to disk
with open('modelos/modelo_random_forest.pkl', 'wb') as pickle_file:
      joblib.dump(modelo_v1, 'modelos/modelo_random_forest.pkl')
```

[211]:
```
# Dictionary with model_v2 metrics
dict_modelo_v2 = {'Nome': 'modelo_randomForest',
                  'Algoritmo': 'Random Forest',
                  'ROC_AUC Score': roc_auc_v2,
                  'AUC Score': auc_v2,
                  'Acurácia': acuracia_v2}
```

[212]:
```
# Add dict to dataframe
df_modelos = df_modelos.append(dict_modelo_v2, ignore_index = True)
```

[213]:
```
display(df_modelos)
```

```
                  Nome            Algoritmo  ROC_AUC Score  AUC Score  \
0            modelo_v1  Regressão Logística       0.698831   0.809892
1  modelo_randomForest        Random Forest       0.767238   0.840602

   Acurácia
0  0.834433
1  0.779844
```

5 Model 3 with KNN

[214]:
```
# List of possible values of K
vizinhos = list(range(1, 20, 2))
```

[215]:
```
# List for the scores
cv_scores = []
```

[216]:
```
# Cross-validation to determine the best value of k
for k in vizinhos:
    knn = KNeighborsClassifier(n_neighbors = k)
    scores = cross_val_score(knn, X_treino, y_treino, cv = 5, scoring =␣
 ↪'accuracy')
    cv_scores.append(scores.mean())
```

[217]:
```
# Adjusting the classification error
erro = [1 - x for x in cv_scores]
```

[218]:
```
# Determining the best value of k (with smallest error)
optimal_k = vizinhos[erro.index(min(erro))]
print('O valor ideal de k é %d' % optimal_k)
```

```
O valor ideal de k é 15
```

[219]:
```
# Create the model version 3
modelo_v3 = KNeighborsClassifier(n_neighbors = optimal_k)
```

[220]:
```
# Model training
modelo_v3.fit(X_treino, y_treino)
```

[220]:
```
KNeighborsClassifier(n_neighbors=15)
```

[221]:
```
# Predictions under test
y_pred_v3 = modelo_v3.predict(X_teste)
```

[222]:
```
# Confusion matrix
confusion_matrix(y_teste, y_pred_v3)
```

[222]:
```
array([[1127,  316],
       [  60,  164]], dtype=int64)
```

[223]:
```
# Positive class probability prediction
y_pred_proba_v3 = modelo_v3.predict_proba(X_teste)[:,1]
```

[224]:
```
# Calculate ROC_AUC on test
roc_auc_v3 = roc_auc_score(y_teste, y_pred_v3)
print(roc_auc_v3)
```

```
0.7565773190773191
```

[225]:
```
# Calculate ROC curve
fpr_v3, tpr_v3, thresholds = roc_curve(y_teste, y_pred_proba_v3)
```

[226]:
```
# AUC in test
auc_v3 = auc(fpr_v3, tpr_v3)
print(auc_v3)
```

```
0.8424057642807643
```

[227]:
```
# Test Accuracy
acuracia_v3 = accuracy_score(y_teste, y_pred_v3)
print(acuracia_v3)
```

```
0.7744451109778044
```

[228]:
```
# Save the template to disk
with open('modelos/modelo_knn.pkl', 'wb') as pickle_file:
      joblib.dump(modelo_v3, 'modelos/modelo_knn.pkl')
```

[229]:
```
# Dictionary with model_v3 metrics
dict_modelo_v3 = {'Nome': 'modelo_knn',
                  'Algoritmo': 'KNN',
                  'ROC_AUC Score': roc_auc_v3,
                  'AUC Score': auc_v3,
                  'Acurácia': acuracia_v3}
```

[230]:
```
# Add dict to dataframe
df_modelos = df_modelos.append(dict_modelo_v3, ignore_index = True)
```

[231]:
```
display(df_modelos)
```

```
                  Nome            Algoritmo  ROC_AUC Score  AUC Score  \
0           modelo_v1  Regressão Logística       0.698831   0.809892
1  modelo_randomForest        Random Forest       0.767238   0.840602
2           modelo_knn                  KNN       0.756577   0.842406

   Acurácia
0  0.834433
1  0.779844
2  0.774445
```

6 Model 4 with Decision Tree

[232]:
```
# Hyperparameters
tuned_params_v4 = {'min_samples_split': [2, 3, 4, 5, 7],
                   'min_samples_leaf': [1, 2, 3, 4, 6],
                   'max_depth': [2, 3, 4, 5, 6, 7]}
```

[233]:
```
# Create the model with RandomizedSearchCV
modelo_v4 = RandomizedSearchCV(DecisionTreeClassifier(),
                               tuned_params_v4,
                               n_iter = 15,
                               scoring = 'roc_auc',
                               n_jobs = -1)
```

[234]:
```
# Model training
modelo_v4.fit(X_treino, y_treino)
```

[234]:
```
RandomizedSearchCV(estimator=DecisionTreeClassifier(), n_iter=15, n_jobs=-1,
                   param_distributions={'max_depth': [2, 3, 4, 5, 6, 7],
                                        'min_samples_leaf': [1, 2, 3, 4, 6],
                                        'min_samples_split': [2, 3, 4, 5, 7]},
                   scoring='roc_auc')
```

[235]:
```
# Extract the best model
modelo_v4.best_estimator_
```

[235]:
```
DecisionTreeClassifier(max_depth=6, min_samples_leaf=4, min_samples_split=4)
```

[236]:
```
# Predictions under test
y_pred_v4 = modelo_v4.predict(X_teste)
```

[237]:
```
# Probability predictions
y_pred_proba_v4 = modelo_v4.predict_proba(X_teste)[:,1]
```

[238]:
```
# Confusion matrix
confusion_matrix(y_teste, y_pred_v4)
```

[238]:
```
array([[1206,  237],
       [  46,  178]], dtype=int64)
```

[239]:
```
# Calculates ROC AUC score
roc_auc_v4 = roc_auc_score(y_teste, y_pred_v4)
print(roc_auc_v4)
```

```
0.8152008464508463
```

[240]:
```
# ROC Curve
fpr_v4, tpr_v4, thresholds = roc_curve(y_teste, y_pred_proba_v4)
```

[241]:
```
# AUC in test
auc_v4 = auc(fpr_v4, tpr_v4)
print(auc_v4)
```

```
0.8523135085635084
```

[242]:
```
# Test Accuracy
acuracia_v4 = accuracy_score(y_teste, y_pred_v4)
print(acuracia_v4)
```

```
0.8302339532093581
```

[243]:
```
# Save the template to disk
with open('modelos/modelo_decision_tree.pkl', 'wb') as pickle_file:
      joblib.dump(modelo_v4, 'modelos/modelo_decision_tree.pkl')
```

[244]:
```
# Dictionary with model_v4 metrics
dict_modelo_v4 = {'Nome': 'modelo_decisionTree',
                  'Algoritmo': 'Decision Tree',
                  'ROC_AUC Score': roc_auc_v4,
                  'AUC Score': auc_v4,
                  'Acurácia': acuracia_v4}
```

[245]:
```
# Add dict to dataframe
df_modelos = df_modelos.append(dict_modelo_v4, ignore_index = True)
```

[246]:
```
display(df_modelos)
```

```
                  Nome            Algoritmo  ROC_AUC Score  AUC Score  \
0            modelo_v1  Regressão Logística       0.698831   0.809892
1  modelo_randomForest        Random Forest       0.767238   0.840602
2           modelo_knn                  KNN       0.756577   0.842406
3  modelo_decisionTree        Decision Tree       0.815201   0.852314

   Acurácia
0  0.834433
1  0.779844
2  0.774445
3  0.830234
```

7 Model 5 with SVM

[247]:
```
# Function for hyperparameter selection
def svc_param_selection(X, y, nfolds):
    Cs = [0.001, 0.01, 0.1, 1, 10]
    gammas = [0.001, 0.01, 0.1, 1]
    param_grid = {'C': Cs, 'gamma' : gammas}
    grid_search = GridSearchCV(SVC(kernel = 'rbf'), param_grid, cv = nfolds)
    grid_search.fit(X_treino, y_treino)
    grid_search.best_params_
    return grid_search.best_params_
```

[248]:
```
# Apply the function
svc_param_selection(X_treino, y_treino, 5)
```

[248]:
```
{'C': 10, 'gamma': 0.1}
```

[249]:
```
# Create the model with the best hyperparameters
modelo_v5 = SVC(C = 1, gamma = 1, probability = True)
```

[250]:
```
# Model training
modelo_v5.fit(X_treino, y_treino)
```

[250]:
```
SVC(C=1, gamma=1, probability=True)
```

[251]:
```
# Predictions under test
y_pred_v5 = modelo_v5.predict(X_teste)
```

[252]:
```
confusion_matrix(y_teste, y_pred_v5)
```

[252]:
```
array([[1209,  234],
       [  57,  167]], dtype=int64)
```

[253]:
```
# Probability predictions
y_pred_proba_v5 = modelo_v5.predict_proba(X_teste)[:, 1]
```

[254]:
```
# Calculates ROC AUC score
roc_auc_v5 = roc_auc_score(y_teste, y_pred_v5)
print(roc_auc_v5)
```

```
0.7916867760617761
```

[255]:
```
# Calculate ROC curve
fpr_v5, tpr_v5, thresholds = roc_curve(y_teste, y_pred_proba_v5)
```

[256]:
```
# AUC in test
auc_v5 = auc(fpr_v5, tpr_v5)
print(auc_v5)
```

```
0.8590795465795465
```

[257]:
```
# Test Accuracy
acuracia_v5 = accuracy_score(y_teste, y_pred_v5)
print(acuracia_v5)
```

```
0.8254349130173965
```

[258]:
```
# Save the template to disk
with open('modelos/modelo_svm.pkl', 'wb') as pickle_file:
      joblib.dump(modelo_v5, 'modelos/modelo_svm.pkl')
```

[259]:
```
# Dictionary with model_v5 metrics
dict_modelo_v5 = {'Nome': 'modelo_svm',
                  'Algoritmo': 'SVM',
```

```
                    'ROC_AUC Score': roc_auc_v5,
                    'AUC Score': auc_v5,
                    'Acurácia': acuracia_v5}
```

[260]:
```
# Add dict to dataframe
df_modelos = df_modelos.append(dict_modelo_v5, ignore_index = True)
```

[261]:
```
display(df_modelos)
```

```
                  Nome            Algoritmo  ROC_AUC Score  AUC Score  \
0            modelo_v1  Regressão Logística       0.698831   0.809892
1  modelo_randomForest        Random Forest       0.767238   0.840602
2           modelo_knn                  KNN       0.756577   0.842406
3  modelo_decisionTree        Decision Tree       0.815201   0.852314
4           modelo_svm                  SVM       0.791687   0.859080

   Acurácia
0  0.834433
1  0.779844
2  0.774445
3  0.830234
4  0.825435
```

7.1 Best Model Selection

[262]:
```
# We will use the model with the highest AUC Score, because it is a global␣
 ↪metric
# The AUC Score is ideal for comparing models from different algorithms
df_melhor_modelo = df_modelos[df_modelos['AUC Score'] == df_modelos['AUC␣
 ↪Score'].max()]
```

[263]:
```
df_modelos['AUC Score'].max()
```

[263]:
```
0.8590795465795465
```

[264]:
```
df_melhor_modelo
```

[264]:
```
         Nome Algoritmo  ROC_AUC Score  AUC Score  Acurácia
4  modelo_svm       SVM       0.791687    0.85908  0.825435
```