Questao_2

February 28, 2021

1 Questão 2

In [6]:
```
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [11]:
```
path = "teste_smarkio_lbs.xls"
df1 = pd.read_excel(path, 'Análise_ML')
```

In [12]:
```
df1.head()
```

Out[12]:

	Pred_class	probabilidade	status	True_class
0	2	0.079892	approved	0.0
1	2	0.379377	approved	74.0
2	2	0.379377	approved	74.0
3	2	0.420930	approved	74.0
4	2	0.607437	approved	NaN

In [13]:
```
#Separando apenas os approved
df_ap = df1[df1['status']=='approved']
```

In [14]:
```
#Substituindo os NaN pela classe correta
df1['True_class'] = np.where(df1['True_class'].isnull(),\
                             df1['Pred_class'],df1['True_class'])
```

In [15]:
```
df1.head()
```

Out[15]:

	Pred_class	probabilidade	status	True_class
0	2	0.079892	approved	0.0
1	2	0.379377	approved	74.0
2	2	0.379377	approved	74.0
3	2	0.420930	approved	74.0
4	2	0.607437	approved	2.0

In [16]:
```
df1['True_class'].isnull().value_counts()
```

Out[16]:
```
False    643
Name: True_class, dtype: int64
```

1.1 Métricas

Considerando:

- VP = verdadeiros positivos
- FP = falsos posistivos
- FN = falsos negativos
- VF = verdadeiros falsos

Vamos utilizar as seguintes métricas:

- A **Acurácia**, que mede o numero de predições corretas em relação ao total de predições ((VP+VF) / (VP+FP+FN+VF))
- A **Precisão**, que quantifica o número de verdadeiros positivos dentre todas as previsões positivas (VP/(VP+FP))
- O **Recall**, que quantifica o numero de verdadeiros positivos dentro todos os exemplos positivos do dataset (VP/(VP+FN))
- O **F1**, que é um média harmonica entre a **Precisão** e o **Recall** (2 * **Precisao** * **Recall** /(**Precisao**+**Recall**))

Uma vez que estamos tratando de um problema de múltiplas classes, vamos considerar as *weighted averages* da **Precisão**, do **Recall** e do **F1**, fazendo assim uma média ponderada dos indicadores por classe. Estas métricas podem ser encontradas abaixo:

```
In [17]: from sklearn import metrics
         print(metrics.classification_report(df1['True_class'],df1['Pred_class']))
```

```
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        54
         2.0       0.77      0.77      0.77        61
         3.0       0.79      0.83      0.81        60
         4.0       0.78      0.86      0.82        21
        11.0       0.44      1.00      0.62         4
        12.0       0.83      0.71      0.77         7
        15.0       0.67      0.67      0.67         3
        17.0       0.86      0.75      0.80         8
        19.0       0.40      0.40      0.40         5
        21.0       0.00      0.00      0.00         0
        22.0       0.67      0.91      0.77        11
        24.0       0.71      0.62      0.67        16
        25.0       0.83      1.00      0.91        10
        26.0       0.33      0.50      0.40         2
        28.0       0.50      1.00      0.67         1
        29.0       1.00      1.00      1.00         7
        30.0       0.60      1.00      0.75         3
        31.0       0.00      0.00      0.00         0
        32.0       0.25      0.50      0.33         4
        33.0       0.00      0.00      0.00         0
```

36.0	1.00	1.00	1.00	1
39.0	0.67	1.00	0.80	4
40.0	1.00	0.78	0.88	9
43.0	0.50	1.00	0.67	3
46.0	1.00	1.00	1.00	1
48.0	0.50	0.25	0.33	4
49.0	0.00	0.00	0.00	0
50.0	0.00	0.00	0.00	0
52.0	0.30	1.00	0.46	6
54.0	1.00	1.00	1.00	2
55.0	0.82	0.93	0.87	15
56.0	1.00	1.00	1.00	3
58.0	1.00	0.50	0.67	2
59.0	0.25	1.00	0.40	1
60.0	0.81	0.89	0.85	28
62.0	0.75	0.60	0.67	5
63.0	1.00	1.00	1.00	2
64.0	0.00	0.00	0.00	0
65.0	0.33	0.50	0.40	2
66.0	0.00	0.00	0.00	1
68.0	0.67	1.00	0.80	2
69.0	1.00	1.00	1.00	1
70.0	1.00	0.75	0.86	4
73.0	0.50	0.50	0.50	2
74.0	0.95	0.72	0.82	78
76.0	1.00	0.80	0.89	10
77.0	0.77	0.83	0.80	29
78.0	1.00	1.00	1.00	3
79.0	1.00	0.43	0.60	7
81.0	0.60	0.33	0.43	9
82.0	1.00	1.00	1.00	5
84.0	1.00	1.00	1.00	1
85.0	0.43	0.60	0.50	10
86.0	0.33	0.20	0.25	5
87.0	1.00	0.67	0.80	6
88.0	1.00	1.00	1.00	3
90.0	0.67	1.00	0.80	2
92.0	0.20	0.50	0.29	2
93.0	1.00	1.00	1.00	1
94.0	1.00	1.00	1.00	1
95.0	0.00	0.00	0.00	0
96.0	0.90	0.90	0.90	21
98.0	1.00	0.71	0.83	7
99.0	0.57	0.80	0.67	10
100.0	1.00	1.00	1.00	1
102.0	1.00	0.62	0.77	8
103.0	0.67	1.00	0.80	6
104.0	0.25	1.00	0.40	1

```
       105.0       0.00      0.00      0.00         0
       106.0       0.25      1.00      0.40         1
       107.0       1.00      1.00      1.00         1
       108.0       0.69      0.90      0.78        10
       109.0       0.00      0.00      0.00         0
       110.0       0.69      1.00      0.81        11
       111.0       0.33      1.00      0.50         1
       112.0       0.50      0.67      0.57         3
       113.0       1.00      1.00      1.00         1
       114.0       1.00      1.00      1.00         1
       115.0       0.80      0.80      0.80         5
       116.0       1.00      0.67      0.80         3
       117.0       0.00      0.00      0.00         3
       118.0       0.40      1.00      0.57         2

    accuracy                            0.72       643
   macro avg       0.63      0.70      0.64       643
weighted avg       0.72      0.72      0.70       643
```

In [ ]: