In [29]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import pickle
from sklearn.linear_model import LogisticRegression

with open ('risco_credito.pkl', 'rb') as f:
    x_risco_credito, y_risco_credito = pickle.load(f)
```

In [19]:
```
x_risco_credito
```

Out[19]:
```
array([[2, 0, 1, 0],
       [1, 0, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 2],
       [1, 1, 1, 2],
       [1, 1, 0, 2],
       [2, 1, 1, 0],
       [2, 1, 0, 2],
       [0, 1, 1, 2],
       [0, 0, 0, 2],
       [0, 0, 1, 0],
       [0, 0, 1, 1],
       [0, 0, 1, 2],
       [2, 0, 1, 1]], dtype=object)
```

In [13]:
```
y_risco_credito # vamos apagar o registro 2,7,11
```

Out[13]:
```
array(['alto', 'alto', 'moderado', 'alto', 'baixo', 'baixo', 'alto',
       'moderado', 'baixo', 'baixo', 'alto', 'moderado', 'baixo', 'alto'],
      dtype=object)
```

In [28]:
```
x_risco_credito = np.delete(x_risco_credito,[2,7], axis = 0) #0 linhas e 1 colunas
y_risco_credito = np.delete(y_risco_credito,[2,7], axis = 0)
```

In [15]:
```
x_risco_credito
```

Out[15]:
```
array([[2, 0, 1, 0],
       [1, 0, 1, 1],
       [1, 1, 1, 2],
       [1, 1, 1, 2],
       [1, 1, 0, 2],
       [2, 1, 1, 0],
       [0, 1, 1, 2],
       [0, 0, 0, 2],
       [0, 0, 1, 0],
       [0, 0, 1, 1],
       [0, 0, 1, 2],
       [2, 0, 1, 1]], dtype=object)
```

In [23]:
```
y_risco_credito
```

Out[23]:
```
array(['alto', 'alto', 'alto', 'baixo', 'baixo', 'alto', 'baixo', 'baixo',
       'alto', 'moderado', 'baixo', 'alto'], dtype=object)
```

In [31]:
```
logistic_risco_credito = LogisticRegression(random_state = 1)
logistic_risco_credito.fit(x_risco_credito, y_risco_credito)
```

Out[31]:
```
LogisticRegression(random_state=1)
```

In [32]:
```
logistic_risco_credito.intercept_
```

Out[32]:
```
array([ 0.55733877, -0.59215165,  0.03481289])
```

In [33]:
```
logistic_risco_credito.coef_
```

Out[33]:
```
array([[ 0.61513831, -0.34196498,  0.49067887, -0.79369827],
       [-0.72760403,  0.135437  , -0.29167359,  0.96796844],
       [ 0.11246572,  0.20652798, -0.19900528, -0.17427017]])
```

In [34]:
```
previsoes1 = logistic_risco_credito.predict([[0,0,1,2],[2,0,0,0]])
previsoes1
```

Out[34]:
```
array(['baixo', 'alto'], dtype=object)
```

In [ ]:
```

```