Exemplo de Uso (Regressão Logística)

Passo 1. Estabeleça um contexto de pesquisa/negócio

Identificar estudantes que estejam desmotivados e agir antes que ocorra evasão.

Passo 2. Obtenha os dados

In [2]:
```
from cdia.datasets import DatasetEstudantes

# Neste caso, utilizamos um framework para a geração de dados
n_amostra = 1000
df = DatasetEstudantes.criar(n_amostra)

# Transforma a motivação em um atributo binário
df.motivação = df.motivação.apply(lambda x: 0 if x < 5 else 1)

# Redução do dataset para atributos com poder preditivo
df = df.filter(['idade', 'renda', 'ano_curso', 'motivação'])

df.head()
```

Out[2]:

	idade	renda	ano_curso	motivação
0	19	391.0	3	1
1	19	217.0	3	1
2	19	164.0	3	1
3	19	66.0	3	1
4	19	249.0	3	1

In [3]:
```
# Variáveis independentes
X = df.filter(['idade', 'renda', 'ano_curso']).to_numpy()

# Variável dependente
y = df.filter(['motivação'])
y = list(y.motivação)
```

Passo 3. Divide o conjunto de treinamento em treino e testes

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

# Divide o conjunto de dados em treino e testes
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    train_size=0.75,
                                                    test_size=0.25,
                                                    random_state=42)
```

Passo 3.1. Treinando o modelo com `sklearn`.

In [5]:
```
# Importa as bibliotecas
from sklearn.linear_model import LogisticRegression
```

In [7]:
```
# Configura os parâmetros
model_params = {
    'random_state': 42,
    'max_iter': 10 ** 4,
}

# Treina um modelo de regressão logística
logreg = LogisticRegression(**model_params)
skl_model = logreg.fit(X_train, y_train)
```

In [8]:
```
# Faz para a previsão para todas instâncias no X_test
skl_model.predict(X_test)
```

Out[8]:
```
array([1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,
       0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 1, 1])
```

In [9]:
```
# Verifica a acurácia
skl_model.score(X_test, y_test)
```

Out[9]:
```
0.956
```

Passo 3.2. Treinando o modelo com `statsmodel`

In [11]:
```
# Importa as bibliotecas
import statsmodels.api as sm
import numpy as np
```

In [13]:
```
sm_logit = sm.Logit(y_train,X_train)
sm_model = sm_logit.fit()
y_pred = sm_model.predict(X_test)
print(sm_model.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.245964
         Iterations 11
                          Results: Logit
==================================================================
Model:              Logit            Pseudo R-squared: 0.619
Dependent Variable: y                AIC:              374.9458
Date:               2022-05-04 08:12 BIC:              388.8060
No. Observations:   750              Log-Likelihood:   -184.47
Df Model:           2                LL-Null:          -484.02
Df Residuals:       747              LLR p-value:      8.1053e-131
Converged:          1.0000           Scale:            1.0000
No. Iterations:     11.0000
------------------------------------------------------------------
       Coef.     Std.Err.       z        P>|z|      [0.025     0.975]
------------------------------------------------------------------
x1     0.4649      0.0448     10.3852     0.0000      0.3771      0.5526
```

```
x2    -0.0037      0.0003    -12.1579    0.0000    -0.0042    -0.0031
x3    -0.1763      0.1328     -1.3276    0.1843    -0.4365     0.0840
==============================================================================

/Users/jeff/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/dis
crete_model.py:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```