Regressão Logística

1. Carregar e explorar os dados

Nessa etapa foi importado os dados e explorado para conhecer suas variáveis e medidas estatísticas

	balance	income
count	10000.000000	10000.000000
mean	835.374886	33516.981876
std	483.714985	13336.639563
min	0.000000	771.967729
25%	481.731105	21340.462903
50%	823.636973	34552.644802
75%	1166.308386	43807.729272
max	2654.322576	73554.233495

Vemos que o conjunto de dados possui 10 000 linhas, com uma média de **fatura** de 835 dólares por mês e uma média de **renda anual** de 33 500 dólares.

Além disso, podemos usar o comando `df.info()` para obter informações a cerca das colunas e tipos de dados do nosso DataSet:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   default  10000 non-null  category
 1   student  10000 non-null  category
 2   balance  10000 non-null  float64
```

```
 3   income   10000 non-null  float64
dtypes: category(2), float64(2)
memory usage: 176.1 KB
```

Ou seja, já é perceptível que temos duas colunas com valores categóricos (colunas **default** e **student**) e vamos ter que transforma-las para valores contínuos ou discretos, que é a próxima etapa na nossa análise.

2. Preparação dos dados

Nessa etapa foi preparado os dados para transformarmos a variável `default` e `student` para binário.

3. Dividir os dados em treino e teste

Foi feita a divisão dos dados conforme pedido no exercício (70% dos dados para treino e 30% para teste).

Ficamos com:

```
Tamanho do dataset de treino:  7000
Tamanho do dataset de teste:  3000
```

Que totaliza 10 000 linhas no total, que era o tamanho de nosso dataset original.

4. Construir o modelo de regressão logística

Foi construída uma função para definirmos o modelo, como mostrado abaixo:

```
def train_model(X_train, X_test, y_train, y_test):
  model = LogisticRegression()
  model.fit(X_train, y_train)
  y_pred = model.predict(X_test)

  accuracy = accuracy_score(y_test, y_pred)
  conf_matrix = confusion_matrix(y_test, y_pred)
```

```
  print("Acurácia: ", accuracy)
  print("Matriz de Confusão:\n", conf_matrix)
  print("\n")

  return y_pred
```

Depois disso, o modelo foi treinado com cada variável de entrada e depois considerando todas as variáveis juntas, e com isso prosseguimos com a análise a seguir.

5. Análise dos coeficientes e interpretação

Obtivemos como resultado dos coeficientes do modelo abaixo, considerando que foi levado em conta o treinamento do modelo com cada variável independente sozinha, e depois considerando todas as variáveis de entrada juntas:

```
Treinando com 'balance'...
Acurácia: 0.9727
Matriz de Confusão:
 [[2894   12]
 [  70   24]]

Coeficiente de intercepto: [-10.8001]
Variável independente: ['balance']
Coeficiente independente: [[0.0056]]
--------------------------------------------------

Treinando com 'student'...
Acurácia: 0.9687
Matriz de Confusão:
 [[2906    0]
 [  94    0]]

Coeficiente de intercepto: [-3.5289]
Variável independente: ['student']
```

```
Coeficiente independente: [[0.53]]
--------------------------------------------------

Treinando com 'income'...
Acurácia: 0.9687
Matriz de Confusão:
 [[2906    0]
 [  94    0]]

Coeficiente de intercepto: [-2.9572]
Variável independente: ['income']
Coeficiente independente: [[-1.2e-05]]
--------------------------------------------------

Treinando com todas as variáveis (balance, student, incom
e)...
Acurácia: 0.9733
Matriz de Confusão:
 [[2895   11]
 [  69   25]]

Coeficiente de intercepto: [-11.1082]
Variável independente: ['student' 'balance' 'income']
Coeficiente independente: [[-4.67459e-01  5.78900e-03  6.0000
0e-06]]
```

A partir dos resultados, já podemos começar nossa análise:

Coeficiente explicativo *balance:* seu valor de 0.56% mostra que ele, sozinho, explica pouco a variável dependente que queremos classificar (a indadimplência). Ou seja, o saldo médio em seu cartão de crédito influencia (pouco) a chance de uma pessoa ser inadimplnte.

Coeficiente explicativo *student:* seu alto valor de 53% mostra que essa característica influencia diretamente uma pessoa ser inadimplente ou não. Ou seja, temos uma variação de 53% observada na inadimplência se a pessoa for um estudante ou não.

Coeficiente explicativo *income:* é o coeficiente que possui o menor valor explicativo sozinho, o que mostra que essa variável não explica bem a variabilidade nos dados de saída (a inadimplência).

6. Avaliação do modelo

A partir dos resultados mostrados na seção anterior, já podemos analisar algumas métricas de avaliação do modelo, como *acurácia* e *matriz de confusão.*

Pensando na matriz de confusão, antes de analisar o resultado, já podemos pensar que o pior caso (imaginando um cenário real) é o modelo ter a pior performance (errar mais) em casos em que a pessoa é inadimplente mas foi prevista como não inadimplente, pois essa situação poderia causar diversos problemas para o banco.

Coeficiente explicativo *balance:*

Classe Prevista / Classe Real	Inadimplente	Não inadimplente
Inadimplente	2894	12
Não inadimplente	70	24

Coeficiente explicativo *student:*

Classe Prevista / Classe Real	Inadimplente	Não inadimplente
Inadimplente	2906	0
Não inadimplente	94	0

Coeficiente explicativo *income:*

Classe Prevista / Classe Real	Inadimplente	Não inadimplente
Inadimplente	2906	0
Não inadimplente	94	0

Coeficientes explicativos *balance*, *student* e *income:*

Classe Prevista / Classe Real	Inadimplente	Não inadimplente
Inadimplente	2895	11
Não inadimplente	69	25

Nos resultados dos treinamentos apresentados, vemos que a menor quantidade de **Falsos Negativos** (modelo previu que não era inadimplente mas era inadimplente) foi justamente no treinamento com todas as variáveis de entrada juntas, e foi nesse treinamento também que a maior **acurácia** foi obtida.

7. Ajuste do limiar de decisão

Ao diminuirmos o limiar de decisão (threshold) e passar de 0.5 para 0.3, a acurácia do modelo treinado para as três variáveis de entrada diminui:

```
Acurácia: 0.9680
Precisão: 0.4875
Matriz de Confusão:
[[2865   41]
 [  55   39]]
```

Se diminuirmos ainda mais o threshold, a acurácia diminui também. É perceptível que, como **abaixamos o threshold**, o modelo preve muitos **mais casos de Falsos Positivos** (preveu que era inadimplente, mas não é), que somam 41 de acordo com a matriz de confusão, o que faz sentido já que o limiar ficou mais baixo para a previsão de inadimplência. Porém, o número de Falsos Negativos diminuiu, o que é um trade-off e uma escolha a se fazer a respeito do limiar e escolhe-lo da melhor forma de acordo com a situação.

8. Relatório e Conclusão

Portanto, concluímos que a variável independente que possui mais explicabilidade sobre a variabilidade da saída do modelo (variável dependente) é a variável ***student***, de acordo com o coeficiente calculado para essa variável. Além disso, foi mostrado também que diminuir o limiar de decisão afeta a acurácia do modelo,

mas pode ajudar para diminuir outros tipos de erros, como Falsos Negativos de acordo com a matriz de correlação mostrada anteriormente