

## ✓ Questão 1

## ✓ Instalando e importando bibliotecas

```
!pip install scikit-multilearn
!pip install scikit-learn
!pip install nltk
!pip install imbalanced-learn
```

```
Requirement already satisfied: scikit-multilearn in /usr/local/lib/python3.11/dist-packages (0.2.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.5.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.11/dist-packages (0.13.0)
Requirement already satisfied: numpy<3,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from imbalanced-learn) (2.0.2)
Requirement already satisfied: scipy<2,>=1.10.1 in /usr/local/lib/python3.11/dist-packages (from imbalanced-learn) (1.15.3)
Requirement already satisfied: scikit-learn<2,>=1.3.2 in /usr/local/lib/python3.11/dist-packages (from imbalanced-learn) (1.6.1)
Requirement already satisfied: sklearn-compat<1,>=0.1 in /usr/local/lib/python3.11/dist-packages (from imbalanced-learn) (0.1.3)
Requirement already satisfied: joblib<2,>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from imbalanced-learn) (1.5.1)
Requirement already satisfied: threadpoolctl<4,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from imbalanced-learn) (3.6.0)
```

```
import pandas as pd
import numpy as np
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, hamming_loss, classification_report
from skmultilearn.problem_transform import BinaryRelevance
from nltk.corpus import stopwords
import nltk
from imblearn.over_sampling import RandomOverSampler
from scipy.sparse import csr_matrix
from scipy.sparse import vstack
```

## ✓ Carregando a base de dados

```
nltk.download('stopwords')

# Carregar o dataset
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
test_labels = pd.read_csv('test_labels.csv')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
```

```
# Visualizando
print("Dados de treino:\n")
print(train.head())
print(train.shape)

print("\nDados de teste:\n")
print(test.head())
print(test.shape)

print("\nLabels de teste:\n")
print(test_labels.head())
print(test_labels.shape)
```

```
Dados de treino:
```

id

comment\_text toxic \

```

0 0000997932d777bf Explanation\nWhy the edits made under my usern... 0
1 000103f0d9cfb60f D'aww! He matches this background colour I'm s... 0
2 000113f07ec002fd Hey man, I'm really not trying to edit war. It... 0
3 0001b41b1c6bb37e "\nMore\nI can't make any real suggestions on ... 0
4 0001d958c54c6e35 You, sir, are my hero. Any chance you remember... 0

severe_toxic  obscene  threat  insult  identity_hate
0             0        0        0        0        0
1             0        0        0        0        0
2             0        0        0        0        0
3             0        0        0        0        0
4             0        0        0        0        0
(159571, 8)

```

Dados de teste:

```

id comment_text
0 00001cee341fdb12 Yo bitch Ja Rule is more succesful then you'lll...
1 0000247867823ef7 == From RfC == \n\n The title is fine as it is...
2 00013b17ad220c46 " \n\n == Sources == \n\n * Zawe Ashton on Lap...
3 00017563c3f7919a :If you have a look back at the source, the in...
4 00017695ad8997eb I don't anonymously edit articles at all.
(153164, 2)

```

Labels de teste:

```

id toxic severe_toxic obscene threat insult \
0 00001cee341fdb12 -1 -1 -1 -1 -1
1 0000247867823ef7 -1 -1 -1 -1 -1
2 00013b17ad220c46 -1 -1 -1 -1 -1
3 00017563c3f7919a -1 -1 -1 -1 -1
4 00017695ad8997eb -1 -1 -1 -1 -1

identity_hate
0 -1
1 -1
2 -1
3 -1
4 -1
(153164, 7)

```

## ✓ Pré-processamento do texto

```

# Labels de classificação
labels = ['toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate']

```

```

# Pré-processamento dos textos
def preprocess(text):
    text = str(text).lower()
    text = re.sub(r'[\w\s]', '', text) # Remove pontuação
    text = re.sub(r'\d+', '', text) # Remove números
    stop_words = set(stopwords.words('english'))
    text = " ".join([word for word in text.split() if word not in stop_words])
    return text

```

```

# Aplicar pré-processamento
train['comment_text'] = train['comment_text'].apply(preprocess)
test['comment_text'] = test['comment_text'].apply(preprocess)

```

```

# Vetorização TF-IDF
tfidf = TfidfVectorizer(max_features=10000)
X_train = tfidf.fit_transform(train['comment_text'])
X_test = tfidf.transform(test['comment_text'])

```

```

y_train = train[labels].values
y_test_labels = test_labels[labels].values

```

```

# Filtrar apenas os exemplos válidos no conjunto de teste
# (Aqueles que não têm -1 em nenhuma das labels)
valid_rows = ~(y_test_labels == -1).any(axis=1)

```

```

X_test_valid = X_test[valid_rows]
y_test_valid = y_test_labels[valid_rows]

```

## ✓ Modelo 1: Sem balanceamento


```

model_base = BinaryRelevance(classifier=LogisticRegression(max_iter=1000))
model_base.fit(X_train, y_train)
y_pred_base = model_base.predict(X_test_valid)

```

## ▼ Avaliação do Modelo 1

```
print("\n=== Avaliação Modelo Base ===")
print("Hamming Loss:", hamming_loss(y_test_valid, y_pred_base))
print("F1 Micro:", f1_score(y_test_valid, y_pred_base, average='micro'))
print("F1 Macro:", f1_score(y_test_valid, y_pred_base, average='macro'))
print("Subset Accuracy:", accuracy_score(y_test_valid, y_pred_base))
print(classification_report(y_test_valid, y_pred_base, target_names=labels))
```



```
=== Avaliação Modelo Base ===
Hamming Loss: 0.025782300165681952
F1 Micro: 0.6370071520264075
F1 Macro: 0.4871900326087752
Subset Accuracy: 0.8951201975679139
```

	precision	recall	f1-score	support
toxic	0.65	0.71	0.68	6090
severe_toxic	0.41	0.32	0.36	367
obscene	0.75	0.61	0.68	3691
threat	0.45	0.18	0.26	211
insult	0.73	0.52	0.61	3427
identity_hate	0.62	0.24	0.35	712
micro avg	0.68	0.60	0.64	14498
macro avg	0.60	0.43	0.49	14498
weighted avg	0.68	0.60	0.63	14498
samples avg	0.06	0.06	0.06	14498

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined ar
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and t
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: F-score is ill-defined and
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

## ▼ Modelo 2: Com Balanceamento (Oversampling)

```
# Listas para armazenar dados balanceados temporariamente
X_list = []
y_list = []

for i, label in enumerate(labels):
    print(f"\nBalanceando label: {label}")

    # Extraí o rótulo específico
    y_label = y_train[:, i]

    # Inicializa o oversampler
    ros = RandomOverSampler(random_state=42)

    # Aplica o oversampling para essa label específica
    X_res, y_res = ros.fit_resample(X_train, y_label)

    # Cria o vetor de labels multilabel com zeros
    y_res_multi = np.zeros((X_res.shape[0], len(labels)), dtype=int)

    # Preenche apenas a coluna da label atual com os valores balanceados
    y_res_multi[:, i] = y_res

    # Armazena os dados
    X_list.append(X_res)
    y_list.append(y_res_multi)

# Concatena todos os X e Y das labels balanceadas
X_bal = vstack(X_list)
y_bal = np.vstack(y_list)

# Remove duplicatas
# df_X = pd.DataFrame(X_bal.toarray())
# df_y = pd.DataFrame(y_bal, columns=labels)
# df_concat = pd.concat([df_X, df_y], axis=1).drop_duplicates()

# Separa novamente X e y
# X_bal = csr_matrix(df_concat.iloc[:, :-len(labels)].values)
# y_bal = df_concat.iloc[:, -len(labels):].values

print("\nShape final após balanceamento e remoção de duplicatas:", X_bal.shape, y_bal.shape)
```

```
# Modelo balanceado
model_balanced = BinaryRelevance(classifier=LogisticRegression(max_iter=1000))
model_balanced.fit(X_bal, y_bal)

# Predição no conjunto de teste
y_pred_balanced = model_balanced.predict(X_test_valid)
```

▼ Avaliação do Modelo 2

```
print("\n=== Avaliação Modelo Balanceado ===")
print("Hamming Loss:", hamming_loss(y_test_valid, y_pred_balanced))
print("F1 Micro:", f1_score(y_test_valid, y_pred_balanced, average='micro'))
print("F1 Macro:", f1_score(y_test_valid, y_pred_balanced, average='macro'))
print("Subset Accuracy:", accuracy_score(y_test_valid, y_pred_balanced))
print(classification_report(y_test_valid, y_pred_balanced, target_names=labels))
```

▼ Questão 2

Questão	Resposta	Justificativa
1	A	A alternativa apresenta corretamente a contrapositiva da sentença "Se é feriado, os bancos estão fechados", que é "Se os bancos não estão fechados, então não é feriado".
4	E	Utilizando o <b>silogismo disjuntivo</b> , se uma das alternativas "não chover" leva Cláudia à praia e "chover" leva Fábia ao clube, então, independentemente do tempo, uma das duas vai à praia.
5	A	Aplicação do <b>modus ponens</b> : se as premissas são verdadeiras, então a conclusão também é verdadeira.
7	A	Uso do <b>silogismo hipotético</b> , onde se $A$ implica $B$ e $B$ implica $C$ , então $A$ implica $C$ .
8	B	Se não vou à escola, então não há aula. Pela <b>contraposição</b> , se há aula, então vou à escola.
9	E	A alternativa correta aplica a regra do <b>silogismo disjuntivo</b> , eliminando uma opção para concluir a outra.
13	C	A alternativa faz uso da <b>regra de contraposição</b> , que é logicamente equivalente à condicional.
16	E	Pelo <b>silogismo disjuntivo</b> , ao negar uma das alternativas de uma disjunção verdadeira, conclui-se que a outra é verdadeira.
17	B	Uso da <b>adição</b> , que permite que, de uma proposição simples, se derive uma disjunção verdadeira.
19	A	Aplicação direta da regra do <b>modus ponens</b> : se $p \rightarrow q$ e $p$ é verdadeiro, então $q$ também é.
20	B	Pela <b>contraposição</b> , a condicional "Se $p$ então $q$ " é equivalente a "Se não $q$ então não $p$ ".
21	C	Aplicação do <b>silogismo disjuntivo</b> , descartando uma opção para concluir a outra.
23	B	Uso da <b>contraposição</b> , transformando "Se $p$ então $q$ " em "Se não $q$ então não $p$ ".
24	C	Uso da <b>adição</b> , que permite criar uma disjunção a partir de uma única proposição verdadeira.
26	D	Aplicação da regra de <b>contraposição</b> , reconhecendo a equivalência lógica entre uma condicional e sua contrapositiva.
27	C	Pela <b>exportação</b> , uma condicional com conjunção no antecedente pode ser reescrita como uma condicional encadeada.
28	A	Aplicação do <b>modus tollens</b> : se $p \rightarrow q$ e $q$ é falso, então $p$ é falso.
31	E	Não se pode concluir diretamente nada sobre a culpa de Francisco, apenas que não desviou dinheiro. A alternativa E é a que resta como logicamente correta, mas sem informação suficiente para ser concluída.
32	A	Aplicação da <b>contraposição</b> : "Se Rodrigo mentiu então ele é culpado" é equivalente a "Se ele não é culpado então ele não mentiu".

▼ Questão 3

a) Aplicação da regra clássica:

“Se a altura  $\geq 170$  cm, então é alto (1); senão, não é alto (0)”.

Personagem	Altura (cm)	Alto (lógica clássica)
Ana	148	0
Bruno	165	0
Carla	172	1
Diego	180	1
Elisa	191	1

b) Discutindo se faz sentido dizer que alguém com 169 cm não é nada alto (0) e alguém com 170 cm é totalmente alto (1)?

Não faz sentido absoluto. A diferença de 1 cm (169 para 170) não representa uma mudança drástica no conceito de "alto". Por isso, a lógica fuzzy é mais adequada, pois permite graus de pertencimento.

c) Propondo uma função de pertinência fuzzy para a categoria "alto", que varie de 0 (não é alto) até 1 (muito alto), usando a seguinte lógica linear aproximada:

- Se altura  $< 160 \rightarrow$  grau = 0
- Se altura  $> 190 \rightarrow$  grau = 1

- Se altura entre 160 e 190  $\rightarrow$  grau =  $\frac{(altura-160)}{(190-160)}$

Ou seja:

$$\text{grau} = \frac{\text{altura} - 160}{30}$$

d) Preenchendo a tabela com os valores fuzzy de "alto":

Personagem	Altura (cm)	Grau de "alto" (fuzzy)
Ana	148	0
Bruno	165	$\frac{(165-160)}{30} = 0,166... \approx 0,17$
Carla	172	$\frac{(172-160)}{30} \approx 0,4$
Diego	180	$\frac{(180-160)}{30} \approx 0,666... \approx 0,67$
Elisa	191	1