Lista IA – 3

**ALUNO**: Caio Gomes Alcântara Glória

**MATRICULA**: 763989

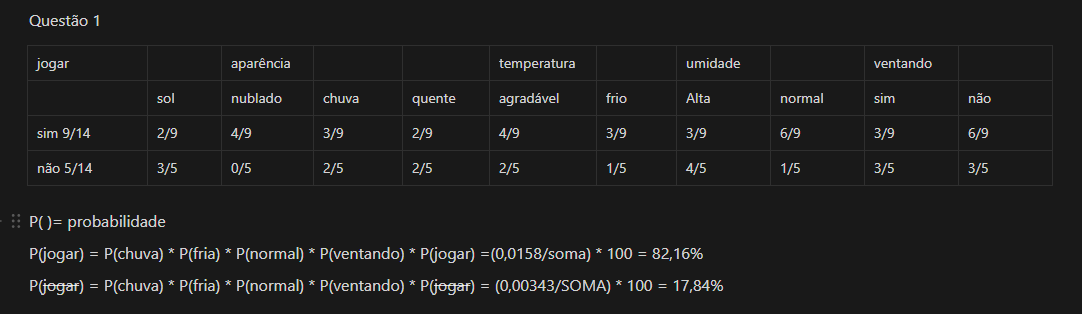
**PROFESSOR**(A): Cristiane Neri

1-

- Total de registros = 14

- Jogou (Sim) = 9

- Não Jogou (Não) = 5



2-

# Importando bibliotecas necessárias

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split, RandomizedSearchCV

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.impute import SimpleImputer

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

from imblearn.over\_sampling import SMOTE

# Carregando a base de dados

df = pd.read\_csv('titanic.csv')

# Visualizando os primeiros registros

print(df.head())

# Etapa 1: Pré-processamento dos dados

# Colunas 'Age', 'Fare' e 'Embarked' possuem valores ausentes

imputer = SimpleImputer(strategy='mean')

df['Age'] = imputer.fit\_transform(df[['Age']])

df['Fare'] = imputer.fit\_transform(df[['Fare']])

df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])

# 1.2 Remover colunas desnecessárias ou redundantes

df = df.drop(columns=['Cabin', 'Ticket', 'Name', 'PassengerId'])

# 1.3 Codificação de variáveis categóricas

label\_encoders = {}

for column in ['Sex', 'Embarked']:

    label\_encoders[column] = LabelEncoder()

    df[column] = label\_encoders[column].fit\_transform(df[column])

# 1.4 Balanceamento dos dados com SMOTE

X = df.drop(columns='Survived')

y = df['Survived']

# Aplicando SMOTE para balancear

smote = SMOTE(random\_state=42)

X\_resampled, y\_resampled = smote.fit\_resample(X, y)

# 1.5

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_resampled, y\_resampled, test\_size=0.2, random\_state=42)

# 1.6 Normalizando

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)