Projeto Integrador Parte C

Alunas

- Gabriella Braz
- Giovana Ribeiro

1) Use o mesmo conjunto de dados já escolhido anteriormente ou escolha um novo conjunto de dados.

```
[96]
df = pd.read_csv("data/tb_1.csv")
print(df.dtypes)
```

```
gender                       object
race_ethnicity               object
parental_level_of_education  object
lunch                        object
test_preparation_course      object
math_score                    int64
reading_score                 int64
writing_score                 int64
dtype: object
```

2) Implemente uma árvore de decisão para classificação.

```
# Projeto Integrador Parte B - Preparação dos Dados

# Entregas:
# 1) Faça um relatório respondendo cada pergunta separadamente.
# 2) Link para a base utilizada.
# 3) Código completo em Python.

# Dando continuidade ao Projeto Integrador - Parte A, faça uma análise dos mesmos
dados utilizados anteriormente, respondendo às seguintes questões:


# ALUNAS
# - Gabriella Braz
# - Giovana Ribeiro

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("data/tb_1.csv")
print(df.dtypes)

df["target"] = (
(df["math_score"] + df["reading_score"] + df["writing_score"]) / 3 >= 60
).astype(int)

X = df.drop("target", axis=1)
y = df["target"]

# Transformar variáveis categóricas em dummies
X = pd.get_dummies(X, drop_first=True)

# Dividir os dados em treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=7)

# Criar e treinar o classificador de árvore de decisão
decision_T = DecisionTreeClassifier(
max_depth=3,
min_samples_split=25,
min_samples_leaf=8,
random_state=7,
)
decision_T.fit(X_train, y_train)

# Visualizar a árvore
plt.figure(figsize=(20, 15)) # Ajustei o tamanho para ficar legível
plot_tree(
decision_T,
filled=True,
feature_names=X.columns,
class_names=df["target"].astype(str).unique(),
)
plt.show()

# Fazer previsões
y_pred = decision_T.predict(X_test)

# Avaliar o modelo
print("Acurácia:", metrics.accuracy_score(y_test, y_pred))


# Importância das variáveis
print("Importância das variáveis:")
```

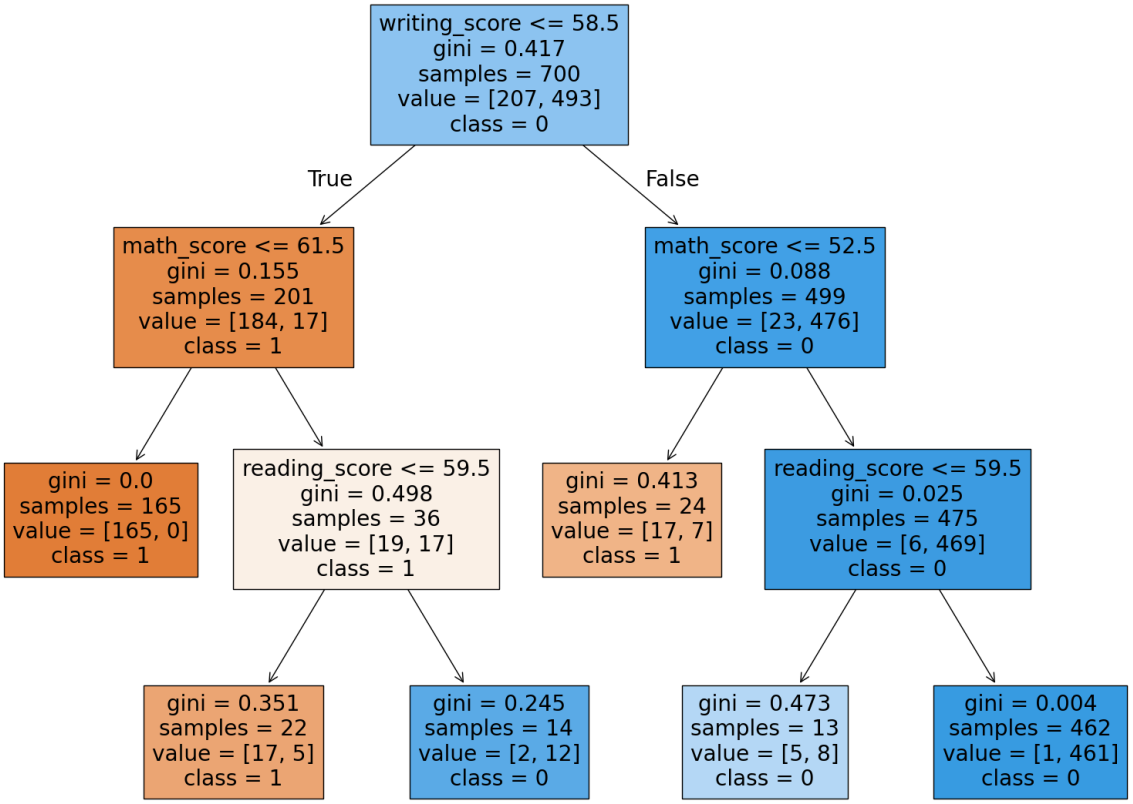
```
importances = decision_T.feature_importances_
for i, feature in enumerate(X.columns):
print(f"- {feature}: {importances[i]:.4f}")

## RELATÓRIO
"""A árvore de decisão treinada para prever se os estudantes teriam média igual ou
superior a 60 apresentou
acurácia de 96,33%, indicando um bom desempenho do modelo. As notas de escrita
(writing_score) foram a
variável mais importante, seguidas por matemática e leitura, enquanto todas as
variáveis categóricas,
como gênero, etnia, nível educacional dos pais, tipo de almoço e curso
preparatório, não tiveram influência
na decisão da árvore.
Esses resultados mostram que, para este conjunto de dados, o desempenho acadêmico
dos alunos é o principal
fator para determinar a aprovação, enquanto características sociodemográficas e de
apoio escolar não impactam
significativamente o modelo."""
```

3) Analise a acurácia.

O modelo de árvore de decisão utilizado para prever a aprovação dos estudantes apresentou uma acurácia de 96,33%, indicando que o modelo classifica corretamente a grande maioria dos casos do conjunto de teste. A análise das variáveis mostrou que o desempenho acadêmico, especialmente a nota de escrita (writing_score), foi o principal fator que influenciou as decisões, enquanto as variáveis categóricas, como gênero, etnia, nível educacional dos pais, tipo de almoço e participação em curso preparatório, não contribuíram para a classificação.

4) Gere a imagem da árvore de decisão e tire insights relevantes para o seu problema.



Writing_score é o fator mais decisivo:

- O nó raiz divide os alunos pelo writing_score <= 58.5. Isso indica que a nota de escrita é o critério mais importante para determinar se o aluno está aprovado ou não. Alunos com writing_score acima de 58.5 têm alta probabilidade de aprovação.

Math_score e Reading_score são critérios secundários:

- Para alunos com writing_score <= 58.5, o math_score (≤ 61.5) é usado para decidir a aprovação.
- Em seguida, o reading_score é usado em alguns ramos menores, indicando que só se torna relevante em casos intermediários.

Variáveis categóricas não aparecem na árvore:

- Nenhuma das variáveis como gender, race_ethnicity, lunch ou test_preparation_course é usada na árvore, reforçando o que os resultados de importância mostraram: o desempenho acadêmico é o principal preditor de aprovação neste dataset.