

Nome: _____ Matrícula: _____
Data: __/__/__

EXERCÍCIO PRÁTICO DE MINERAÇÃO DE DADOS CLASSIFICAÇÃO

Trabalhando com 2 dimensões de dados

Considere a seguinte situação:

- Uma empresa de desenvolvimento de software pretende estimar se um determinado projeto será finalizado ou não com base em 2 dimensões: - Horas estimadas do projeto / - Preço sugerido. Com base nos projetos anteriores, pretende-se estimar se um novo projeto será finalizado ou não. Assim, pode-se reajustar o preço do projeto ou reajustar as horas dedicadas.

unfinished	expected_hours	price
1	26	192

- **unfinished:** indica se foi finalizado ou não. (0 → Finalizado / 1 → Não-Finalizado)
- **expected_hours:** indica as horas estimadas.
- **price:** indica o preço cobrado.

Os dados para treino e testes do modelo encontram-se no arquivo CSV: **projects.csv**

EXERCÍCIO:

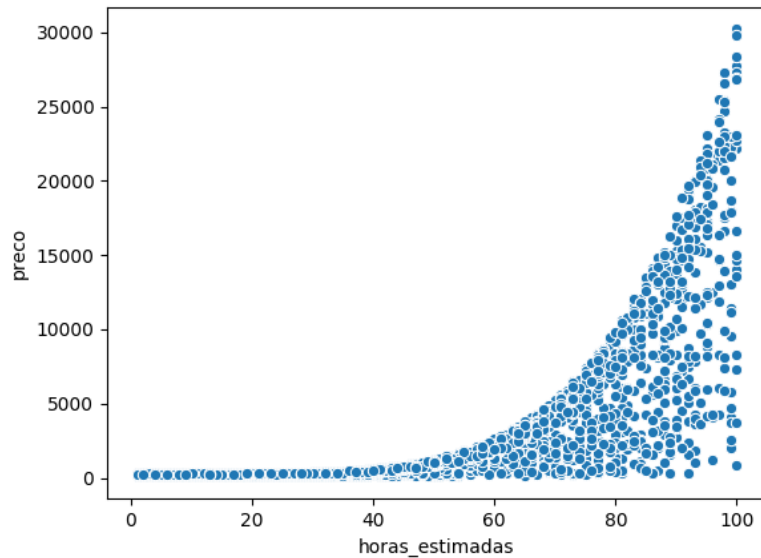
1) Crie um programa em Python que possa ler os dados do arquivo CSV, treinar e aprender com base nos registros e testar possíveis classificações de novos projetos que serão finalizados ou não.

Atividades:

- Trabalhando dados com *dataframe* do Pandas:
 - Utilize a biblioteca **Pandas** para leitura do arquivo CSV.
 - Para ficar mais intuitivo, crie uma nova coluna chamada **finished** com os valores invertidos da coluna **unfinished**. Utilize a função *map* do Pandas.
 - Renomeie as colunas para português utilizando o *rename* do Pandas.
- Visualizando a relação entre as 2 dimensões de dados.
 - Utilize a biblioteca **Seaborn** em conjunto com **Matplotlib** para plotar um gráfico da relação das 2 dimensões. (**GRÁFICO DE DISPERSÃO**)

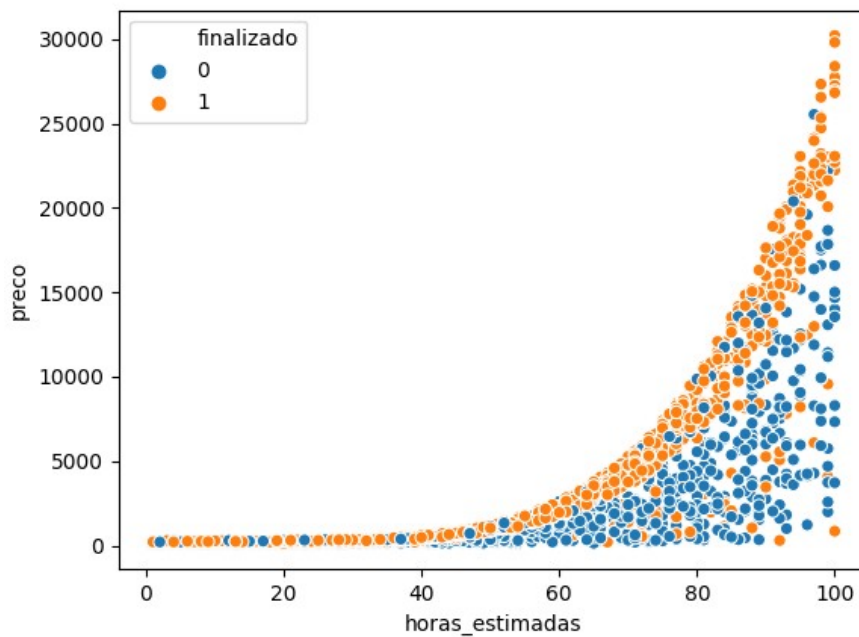
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
ax = sns.scatterplot(x="horas_estimadas", y="preco", data=dados)
plt.show()
```



Agora plote um gráfico que mostre quais dos projetos foram finalizados ou não.

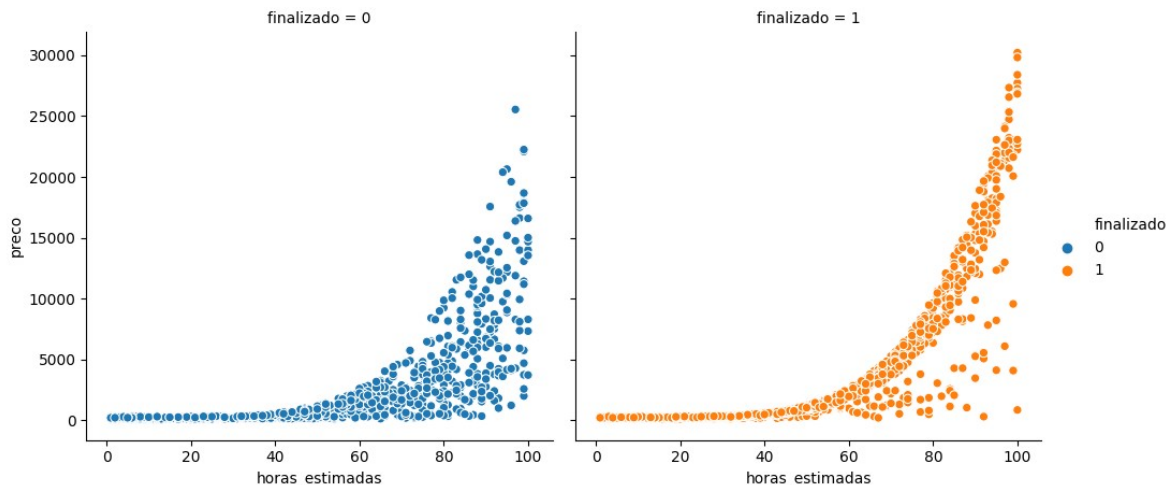
```
ax = sns.scatterplot(x="horas_estimadas", y="preco", hue="finalizado", data=dados)
```



*** Veja que os projetos finalizados que consomem mais horas contém um preço maior.
Esta é a tendência.**

Mesmo gráfico separado:

```
ax = sns.relplot(x="horas_estimadas", y="preco", col="finalizado", hue="finalizado",  
data=dados)
```



*** Veja aqui que existe uma curva padrão para projetos finalizados. Já para os não-finalizados não há curva padrão.**

- Após printar os gráficos, utilize a biblioteca **SKLearn** para treinar e testar a taxa de acertos de projetos que serão finalizados ou não.
 - Para a divisão das bases de treino e teste utilize a função **train_test_split**
 - Utilize 25% da massa de dados para testes.
 - Utilize um **random_test** padrão.
 - Estratifique a porcentagem de Y para as amostras de treino e teste.
- Crie um menu interativo para que o usuário possa entrar com dados de horas estimadas e preço com o resultado se o projeto será finalizado ou não.

Encontre o *baseline*, ou seja, a classificação boba, para usar como parâmetro de avaliação de teste de precisão do algoritmo. Utilize a classe *DummyClassifier* com constante 1 para teste.

- Dentre os algoritmos abaixo, verifique qual algoritmo de classificação apresentará melhor taxa de acerto:

```
from sklearn.dummy import DummyClassifier  
from sklearn.svm import LinearSVC  
from sklearn.svm import SVC  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.linear_model import LinearRegression  
from sklearn.linear_model import LogisticRegression  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.naive_bayes import MultinomialNB  
from sklearn.naive_bayes import GaussianNB  
from sklearn.tree import DecisionTreeClassifier
```

Taxa de acerto para o algoritmo:

A taxa de acerto para o teste foi de 52.22%

Menu interativo para previsões para o usuário:

```
####EFETUANDO PREVISÕES PARA O USUARIO####  
  
Entre com horas estimadas do projeto:  
80  
  
Entre com preço estimado do projeto:  
1000  
  
RESULTADO: Este projeto NÃO será finalizado.  
[0]  
  
...Pressione qualquer tecla...
```

- **Todas *libs* precisam ser instaladas pelo *pip install*
- **Instale a biblioteca *python3-tk* ou *python-tk*.