Introdução

Olá, seja bem-vinda e bem-vindo ao notebook da **aula 05!** Essa será nossa última aula no #Quaren isso chegou a hora de criar seu primeiro modelo de **Machine Learning** em nosso curso. Nessa aul debates importantes na videoaula, então **acompanhar as explicações por meio dela será importa desenvolvimento**.

Antes de começar a aula 5 execute os códigos da aula 04, pois será necessário para essa aula. Fe até o índice Alura 5 e bons estudos!

Aula 4

Aqui iremos explorar e conhecer uma pequena amostra da base de dados do **ENEM 2018**. Esse se passo para construir os **modelos de machine learning da aula 05**. Se você quiser estudar o códig criar o dataset desta aula, pode acessar este **link aqui**.

Vamos iniciar setando a precisão de casas decimais que o pandas irá exibir os dados (`pd.options.display.float_format`), depois lendo e conhecendo as informações contidas na bas

```
import pandas as pd

%precision %.2f
pd.options.display.float_format = '{:,.2f}'.format

uri = "https://github.com/guilhermesilveira/enem-2018/blob/master/MICRODADOS_ENEM_2018_SAM
dados = pd.read_csv(uri)
dados.head()
```

	NU_INSCRICAO	NU_ANO	CO_MUNICIPIO_RESIDENCIA	NO_MUNICIPIO_RESIDENCIA	CO_UF_RES
0	180009327796	2018	2112209	Timon	
1	180012890374	2018	3557105	Votuporanga	
2	180008223824	2018	2506301	Guarabira	
3	180007859645	2018	3534807	Ouro Verde	
4	180013499517	2018	3523107	Itaquaquecetuba	

5 rows × 137 columns

Conheça todas as colunas do nosso dataframe:

```
print(dados.columns.values)
```

Na videoaula tivemos uma discussão muito bacana sobre uma visão geral dos dados e sua organ
ética na IA. Se você não assistiu eu recomendo que veja, não cabe aqui no notebook reproduzir a
vamos seguir com o desenvolvimento.

Conhecidas as informações, vamos chamar o **describe()** para analisá-las. Se atente ao detalhe qu
gera informação de dados numéricos!

```
dados.describe()
```

A saída do `describe` traz várias estatísticas, de forma que algumas não fazem sentido ou não nos momento. Entretanto, se você analisou as últimas colunas, viu que lá temos alguns dados relevan provas e redação.

Desafio você a entrar nos detalhes das análises de notas e diversos outros campos! Como nosso restrito, vamos analisar apenas as notas entre si, mas reflita: Será que existe uma correlação entre tira notas maiores em redação também vai bem em linguagens?

Vamos analisar!

```
colunas_de_notas = ['NU_NOTA_CN', 'NU_NOTA_CH', 'NU_NOTA_LC', 'NU_NOTA_MT', 'NU_NOTA_REDAC
dados_notas = dados[colunas_de_notas].dropna()
dados_notas.columns = ['ciencias_naturais', 'ciencias_humanas', 'linguagem_codigo', 'matem
dados_notas.head()
```

```
len(dados_notas)
```

Como queremos analisar as notas detalhadamente, no código acima separamos apenas os dados Também removemos todos os valores vazios com o `.dropna()` e trocamos os nomes das coluna legível.

Por fim, agora nosso DF tem 97270 linhas e 5 colunas.

Agora sim, vamos calcular a correlação usando o `.corr()`:

```
corr = dados_notas.corr()
corr
```

Temos vários resultados interessantes por aqui: o primeiro é uma correlação considerável entre **li ciencias_humanas**, o que parece fazer sentido. Uma correlação que surpreende é entre **linguagem redacao**. Embora haja uma correlação maior em relação às outras matérias e redação, eu esperav maior do que o existente.

**Mais alguma correlação te chama a atenção? **

Eu tenho mais uma análise das correlações em geral! Repare que as correlações com linguagem_ são as maiores e isso me faz pensar em várias hipóteses!

Será que se eu estudar mais português vou ter um desempenho melhor nas outras matérias? (lem ENEM é uma prova que demanda interpretação de texto, então essa prerrogativa pode fazer senti considerar provas de anos anteriores e comparar as correlações com linguagem_códigos elas ser

A verdade é que uma simples análise de correlação nos gera diversas hipóteses. **Se tiver curiosid essas análises fica como um desafio extra!**

Na videoaula você viu que tentamos plotar diversos gráficos para visualizar a correlação de uma r Abaixo seguem os códigos usados:

```
from string import ascii_letters
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=np.bool))

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

```
sns.heatmap(corr)
```

Depois de apanhar tentando criar boas imagens, resolvemos deixar esse desafio para você kkkkk. aquela cornetada em nosso time uahuahha...

Ok, nós analisamos e conhecemos a base de dados, mas no final o que vou querer é construir um para fazer as predições de algumas notas. Para criar esse modelo de machine learning devemos a distribuição dos nossos dados e verificar se existe alguma tendência entre eles, facilitando o proc

Então, vamos ao **pairplot**:

```
sns.pairplot(dados_notas)
```

Embora existam alguns dados com maior dispersão, outros parecem obedecer uma certa tendênc
desenvolver um modelo de ML com resultados razoáveis será complexo, porém possível (para de
acompanhe a discussão na videoaula).

Com isso chegamos ao final de mais uma aula da #quarentenadados. E aí, o que está achando?

Agora iremos entrar em uma área totalmente nova: o desenvolvimento de modelos de machine le
você esteja empolgado(a) para conhecer um pouquinho mais sobre esse assunto!

Crie seu próprio notebook, reproduza nossa aula e resolva os desafios que deixamos para você.

Até a próxima aula!

P.S: A partir de agora você irá colocar a mão na massa, nossos desafios serão mais analítcos. Que
vivencie o dia-a-dia de um ciêntista de dados, discutindo suas conclusões no Slack e estudando a
outros colegas, por isso não haverá gabarito.

Desafio 6 do Allan Spadini

Estudar o que as pessoas que estudam o assunto estão discutindo e conclusões que já chegaram
de informações (principalmente sensíveis) para machine learning e data science. Podcast do data
sobre o assunto.

Aula 5

Nessa aula discutiremos o que é o processo de classificação e como as máquinas podem aprend
Após essa discussão iniciaremos o tratamento dos dados para criar nosso primeiro modelo de M

A primeira coisa que devemos fazer é separar os dados que vamos usar como entrada do nosso r
precisamos prever:

```
x_4_notas = dados_notas[['ciencias_naturais', 'ciencias_humanas', 'matematica', 'redacao']]
x_4_notas.head()
```

	ciencias_naturais	ciencias_humanas	matematica	redacao
1	523.10	671.30	738.00	680.00
2	409.20	531.70	438.10	600.00
3	452.30	505.50	544.90	400.00
6	428.50	505.20	436.80	560.00
8	491.80	575.70	487.60	660.00

Vamos usar as notas das provas de ciências naturais, ciências humanas, matemática e redação p
da prova de linguagem e códigos.

Como separamos os dados de entrada, também devemos fazer o mesmo com aqueles que precis

```
y_adivinhar = dados_notas['linguagem_codigo']
y_adivinhar
```

Agora temos os dados que precisamos classificar, mas repare que essa é toda nossa informação modelo com todos esses dados, como eu vou conseguir medir a qualidade do modelo?

Por isso precisamos dividir nossos dados em dois grupos, um para treino e outro para teste.

Para fazer isso vamos usar métodos da biblioteca Scikit-learn, ela é uma das principais ferrament Machine Learning! Vale conferir e estudar um pouco mais sobre ela, aqui está o link para a docum

Além do Sklearn, iremos utilizar o numpy para setar o seed dos números pseudo-aleatórios.

```
from sklearn.model_selection import train_test_split
import numpy as np

np.random.seed(43267)

# f(x) = y
x_treino, x_teste, y_treino, y_teste = train_test_split(x_4_notas, y_adivinhar)

print(x_treino.shape)
print(x_teste.shape)
print(y_treino.shape)
print(y_teste.shape)
```

```
x_treino.head()
```

Feita a divisão dos nossos dados, chegou a hora de criar seu primeiro modelo de Regressão(Em a diferença entre regressão e classificação).

Vamos utilizar o LinearSVR do scikit-learn:

```
from sklearn.svm import LinearSVR

modelo = LinearSVR()
modelo.fit(x_treino, y_treino)
```

Até o momento nós treinamos o modelo apenas com o `.fit()`, mas falta fazer a predição dos re realizar a predição chamamos o método `.predict()` do **modelo**.

```
predicoes_notas_linguagem = modelo.predict(x_teste)
predicoes_notas_linguagem[:5]
```

Compare a saída da predição com os valores reais logo abaixo, parece que está fazendo sentido,

```
y_teste[:5]
```

Nos próximos trechos de códigos vamos plotar alguns gráficos! As discussões e todas as análise visualizações foram feitas de forma muito rica na videoaula, portanto recomendo fortemente aco

```
import matplotlib.pyplot as plt

plt.figure(figsize=(9,9))
sns.scatterplot(x=y_teste.values, y=predicoes_notas_linguagem)
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(9,9))
sns.scatterplot(x=y_teste.values, y=y_teste.values - predicoes_notas_linguagem)
```

```
import matplotlib.pyplot as plt

# minha predição TOSCA. Dummy!
plt.figure(figsize=(9,9))
sns.scatterplot(x=y_teste.values, y=y_teste - x_teste.mean(axis=1))
```

```
import matplotlib.pyplot as plt

# predição do paulo TOSCA. Dummy!
plt.figure(figsize=(9,9))
sns.scatterplot(x=y_teste.values, y=y_teste - y_treino.mean())
```

Após discutir esses gráficos, vamos criar mais um modelo de **machine learning** basededo em "árv

```
from sklearn.tree import DecisionTreeRegressor

modelo = DecisionTreeRegressor()
modelo.fit(x_treino, y_treino)
predicoes_notas_linguagem = modelo.predict(x_teste)
plt.figure(figsize=(9,9))
sns.scatterplot(x=y_teste.values, y=predicoes_notas_linguagem)
```

```
plt.figure(figsize=(9,9))
sns.scatterplot(x=x_teste['matematica'].values, y=predicoes_notas_linguagem)
sns.scatterplot(x=x_teste['matematica'].values, y=y_teste.values)
```

Após treinar o modelo e fazer as predições, plotamos duas imagens. A primeira é muito parecida do primeiro classificador, mas a segunda mostra os valores reais e valores previstos!

Essa figura é muito interessante e mostra uma sobreposição muito boa entre elas, indicando que fazem sentido.

Avaliar os modelos por imagens é uma forma relevante, mas não resume a informação muito bem complexo avaliar dois ou três modelos apenas com gráficos.

O que precisamos agora é de uma métrica capaz de nos dizer como nosso modelo está indo, aqu
quadrático médio. Existem centenas de métricas de avaliação, tudo vai depender do que você pre
está prevendo.

```
from sklearn.metrics import mean_squared_error

mean_squared_error(y_teste, predicoes_notas_linguagem)
```

Veja que nosso erro quadrático médio deu em torno dos 4186.22. Embora pelo gráfico nosso moc
bom, pela métrica parece ser um pouco alto.

O MSE, sigla em inglês para essá métrica, é uma medida que quanto mais perto de zero melhor. V
quando calculamos o MSE de dois vetores iguais:

```
mean_squared_error(y_teste, y_teste)
```

Nosso resultado é zero! Você deve estar se perguntando: meu modelo não está nem perto de zerc
ruim assim?

Nós ainda não temos como te dar essa resposta, precisamos de um critério comparativo, pois ass
dizer como nosso modelo está indo. Por exemplo, que tal classificar os nossos dados de uma ma
Para isso temos os chamados métodos **Dummy**.

```
from sklearn.dummy import DummyRegressor

modelo_dummy = DummyRegressor()
modelo_dummy.fit(x_treino, y_treino)
dummy_predicoes = modelo_dummy.predict(x_teste)

mean_squared_error(y_teste, dummy_predicoes)
```

Finalmente conseguimos responder se nosso modelo é tão ruim assim! Na realidade nosso mode
melhores, temos muito o que melhorar, mas já somos melhores que uma classificação ingênua.

Com isso, encerramos nossa última aula. Espero que vocês tenham gostado!

Participem também do nosso **desafio final, valendo um Nintendo Switch**.

Bons estudos e boa sorte!

Forte abraço!

Desafio 1 da Allan Spadini

Explore os parâmetros C e o max_iter do modelo LinearSVR. Não há garantias que o resultado será

Desafio 2 do Thiago Gonçalves

No gráfico em que plotamos a média com o valor previsto, plote a média das 4 notas ao invés de u

Desafio 3 do Paulo Silveira

Remover as notas zero e testar os mesmos modelos, comparando o resultado

Desafio 4 do Guilherme Silveira

Interpretar tudo que foi feito e compartilhar suas conclusões

Desafio 5 do Thiago Gonçalves

Calcule as métricas de erro que utilizamos (mean square root error) também no conjunto de treino acontece comparado com o conjunto de teste.

Não esqueça de compartilhar a solução dos seus desafios c instrutores, seja no Twitter, seja LinkedIn. Boa sorte!