escola
britânica de
artes criativas
& tecnologia

Python para análise de dados

escola britânica de artes criativas & tecnologia

ANÁLISE DE DADOS

GUIA DA AULA 2

Explore dados

- **Explorar dados de créditos**
- **Estrutura**
- **Schema**
- **Dados faltantes**

Acompanhe aqui os temas que serão tratados na videoaula

1. Explorar dados de créditos

Vamos explorar dados de crédito presentes neste neste link:

https://raw.githubusercontent.com/andre-marcos-perez/ebac-course-utils/develop/dataset/credito.csv

Os dados estão no formato CSV e contém informações sobre clientes de uma instituição financeira. Em especial, estamos interessados em explicar a segunda coluna, chamada de **default**, que indica se um cliente é adimplente (`default = 0`) ou inadimplente (`default = 1`) ou seja, queremos entender o motivo de um cliente deixar de honrar com suas dívidas baseado no comportamento de outros atributos, como salário, escolaridade e movimentação financeira. Uma descrição completa dos atributos está a seguir.

1. Explorar dados de créditos

O atributo de interesse (default) é conhecido como **variável resposta** ou **variável dependente**, já os demais atributos que buscam explicá-la (idade, salário etc.) são conhecidas como **variáveis explicativas**, **variáveis independentes** ou **até variáveis preditoras**.

1. Explorar dados de créditos

Coluna	Descrição
id	Número da conta
default	Indica se o cliente é adimplente (0) ou inadimplente (1)
idade	---
sexo	---
depedentes	---
escolaridade	---
estado_civil	---
salario_anual	Faixa do salario mensal multiplicado por 12
tipo_cartao	Categoria do cartao: blue, silver, gold e platinium
meses_de_relacionamento	Quantidade de meses desde a abertura da conta
qtd_produtos	Quantidade de produtos contratados

1. Explorar dados de créditos

Coluna	Descrição
id	Número da conta
iteracoes_12m	Quantidade de iterações com o cliente no último ano
meses_inatico_12m	Quantidade de meses que o cliente ficou inativo no último ano
limite_credito	Valor do limite do cartão de crédito
valor_transacoes_12m	Soma total do valor das transações no cartão de crédito no último ano
qtd_transacoes_12m	Quantidade total de transações no cartão de crédito no último ano

1. Explorar dados de créditos

Vamos começar lendos os dados num dataframe `pandas`.

In [ ]:
```
import pandas as pd
```

In [ ]:
```
df = pd.read_csv(
        'https://raw.githubusercontent.com/andre-marcos-perez/' +  \ 'ebac-course-
        utils/develop/dataset/credito.csv', na_values='na'
)
```

In [ ]:
```
df.head(n=10)
```

Com o dados em mãos, vamos conhecer um pouco melhor a estrutura do nosso conjunto de dados.

2. Estrutura

In [ ]:
```
df.shape # retorna uma tupla (qtd linhas, qtd colunas)
```

In [ ]:
```
df[df['default'] == 0].shape
```

In [ ]:
```
df[df['default'] == 1].shape
```

In [ ]:
```
qtd_total, _ = df.shape
qtd_adimplentes, _ = df[df['default'] == 0].shape qtd_inadimplentes, _ =
df[df['default'] == 1].shape
```

In [ ]:
```
print(f"A proporcão clientes adimplentes é de " + \ f"{round(100 *
        qtd_adimplentes / qtd_total, 2)}%"
      )

print(f"A proporcão clientes inadimplentes é de " + \ f"{round(100 *
        qtd_inadimplentes / qtd_total, 2)}%"
      )
```

3. Schema

In [ ]:
```
df.head(n=5)
```

Colunas e seus respectivos tipos de dados:

In [ ]:
```
df.dtypes
```

Atributos **categóricos**:

In [ ]:
```
df.select_dtypes('object').describe().transpose()
```

Atributos **numéricos**:

In [ ]:
```
df.drop('id',   axis=1).select_dtypes('number').describe().transpose()
```

4. Dados faltantes

Podem ser:

Vazios ("");

Nulos (None);

Não disponíveis ou aplicaveis (na , NA etc.);

Não numérico (nan , NaN , NAN etc).

In [ ]:
```
df.head()
```

Podemos verificar quais colunas possuem dados faltantes.

In [ ]:
```
df.isna().any()
```

4. Dados faltantes

A função a seguir levanta algumas estatísticas sobre as colunas dos dados faltantes:

In [ ]:

```
def  stats_dados_faltantes(df: pd.DataFrame) ->  None:

    stats_dados_faltantes =  []
    for col in df.columns:
        if df[col].isna().any():
            qtd, _ = df[df[col].isna()].shape total, _ =
            df.shape dict_dados_faltantes = {col:
                {
                    'quantidade': qtd,
                    "porcentagem": round(100 *  qtd/total, 2)
                }
            }
            stats_dados_faltantes.append(dict_dados_faltantes)

    for stat in stats_dados_faltantes: print(stat)
```

4. Dados faltantes

In [ ]:
```
stats_dados_faltantes(df=df)
```

In [ ]:
```
stats_dados_faltantes(df=df[df['default'] == 0])
```

In [ ]:
```
stats_dados_faltantes(df=df[df['default'] == 1])
```