escola
britânica de
artes criativas
& tecnologia

Python para análise de dados

escola britânica de artes criativas & tecnologia

ANÁLISE DE DADOS

GUIA DA AULA 3

Transforme e limpe dados

- **Data wrangling**
- **Correção de schema**
- **Remoção de dados faltantes**

Acompanhe aqui os temas que serão tratados na videoaula

1. Data wrangling

Agora que conhecemos melhor a natureza do nosso conjunto de dados, vamos conduzir uma atividade conhecida como *data wrangling* que consiste na transformação e limpeza dos dados do conjunto para que possam ser melhor analisados.

2. Correção de schema

Na etapa de exploração, notamos que as colunas limite_credito e valor_transacoes_12m estavam sendo interpretadas como colunas categóricas (`dtype = object`)

In [ ]:
```
df[['limite_credito', 'valor_transacoes_12m']].dtypes
```

In [ ]:
```
df[['limite_credito',  'valor_transacoes_12m']].head(n=5)
```

Vamos criar uma função lambda para limpar os dados. Mas antes, vamos testar sua aplicação através do método funcional map a seguir.

2. Correção de schema

In [ ]:

```
fn = lambda valor: float(valor.replace(".", "").replace(",", "."))

valores_originais = [
        '12.691,51',
        '8.256,96',
        '3.418,56',
        '3.313,03',
        '4.716,22'
]

valores_limpos = list(map(fn, valores_originais))

print(valores_originais)
print(valores_limpos)
```

2. Correção de schema

Com a função lambda de limpeza pronta, basta aplicá-la nas colunas de interesse.

In [ ]:
```
df['valor_transacoes_12m'] =  df['valor_transacoes_12m'].apply(fn) df['limite_credito'] =
df['limite_credito'].apply(fn)
```

Vamos descrever novamente o *schema*:

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()
```

3. Remoção de dados faltantes

Como o pandas está ciente do que é um dado faltante, a remoção das linhas problemáticas é trivial.

In [ ]:
```
df.dropna(inplace=True)
```

Vamos analisar a estrutura dos dados novamente:

In [ ]:
```
df.shape
```

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

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

3. Remoção de dados faltantes

In [ ]:

```
qtd_total_novo, _ = df.shape
qtd_adimplentes_novo, _ = df[df['default'] == 0].shape
qtd_inadimplentes_novo, _ = df[df['default'] == 1].shape
```

In [ ]:

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

print(f"A nova proporcão de clientes adimplentes é de " + \
        f"{round(100 * qtd_adimplentes_novo / qtd_total_novo, 2)}%"
      )

print("")

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

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