aprendizado de máquina

7.1 - O que é aprendizado de máquina
VIDEO · 13 MIN

7.2 - Preparação dos dados (parte 1)
VIDEO · 19 MIN

7.2 - Preparação dos dados (parte 2)
VIDEO · 10 MIN

7.2 - Preparação dos dados (parte 3)
VIDEO · 12 MIN

Exercício de Código | Preparação dos dados
TEXT

Exercício de revisão | Preparação dos dados
QUIZ · 1 PERGUNTA

7.3 - Introdução a regressão linear (parte 1)
VIDEO · 10 MIN

7.3 - Introdução a regressão linear (parte 2)
VIDEO · 16 MIN

Exercício de Código | Introdução a regressão linear
TEXT

[Módulo 7] Chegou a hora do feedback!
QUIZ · 2 PERGUNTAS · ESSE É PRÉ REQUISITO

Exercício de Código | Preparação dos dados

3 COMENTÁRIOS/DÚVIDAS

📌 **Exercício de Código**

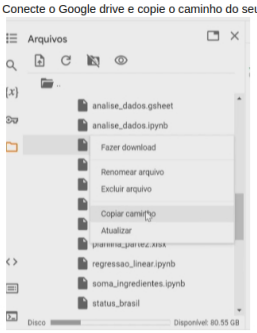
Vamos preparar os nossos dados focando em feature engineering, assim como nossa instrutora!

Acompanhe o gabarito:

Parte 1:

Baixe o arquivo e suba na sua pastinha do drive: analise_dados_mod7 (1).xlsx
Abra um novo arquivo no Google Collaboratory, nomeie como regressao_linear

Conecte o Google drive e copie o caminho do seu arquivo e cole dentro das aspas simples:




```
import pandas as pd
```

```
dados = pd.read_excel('/content/drive/MyDrive/programaria/analise_dados_mod7.xlsx')
```

Vamos filtrar a *situação atual de trabalho*:

```
dados = dados[dados['QUAL SUA SITUAÇÃO ATUAL DE TRABALHO?']=='Empregado (CLT)']
```

```
dados['QUAL SUA SITUAÇÃO ATUAL DE TRABALHO?'].value_counts()
```

Vamos organizar algumas colunas de *cor/raça/etnia* para não atrapalhar o resultado do nosso modelo:

```
lista_retirar = ['Prefiro não informar', 'Outra',  'Indígena']
```

```
dados = dados[~dados['COR/RACA/ETNIA'].isin(lista_retirar)]
```

```
dados['NAO_BRANCA'] = dados['COR/RACA/ETNIA'].apply(lambda x: 1 if x!= 'Branca' else 0)
```

```
dados['TEMPO_EXPERIENCIA'] = dados['QUANTO TEMPO DE EXPERIÊNCIA NA ÁREA DE DADOS VOCÊ TEM?'].str.extract(r'(\d+)')
```

Na coluna *tempo de experiência:*

```
dados['TEMPO_EXPERIENCIA'] = dados['QUANTO TEMPO DE EXPERIÊNCIA NA ÁREA DE DADOS VOCÊ TEM?'].str.extract(r'(\d+)')
```

Na coluna *número de funcionários:*

```
dados['NUMERO DE FUNCIONARIOS'] = dados['NUMERO DE FUNCIONARIOS'].str.replace('.', '')
```

```
dados['NUMERO DE FUNCIONARIOS'] = dados['NUMERO DE FUNCIONARIOS'].str.extract(r'(\d+)')
```

```
dados['TEMPO_EXPERIENCIA'] = dados['TEMPO_EXPERIENCIA'].fillna(0)
```

Parte 2:

Vamos criar nossa coluna insatisfação:

```
dados.loc[dados['Qual o principal motivo da sua insatisfação com a empresa atual?'].notnull(),
'Qual o principal motivo da sua insatisfação com a empresa atual?'].apply(lambda x: 1 if 'Salário' in x else 0)
```

Após rodar, atribua a coluna 'INSATISFACAO' .

```
dados.loc[dados['Qual o principal motivo da sua insatisfação com a empresa atual?'].notnull(), 'INSATISFACAO'] =
dados.loc[dados['Qual o principal motivo da sua insatisfação com a empresa atual?'].notnull(),
'Qual o principal motivo da sua insatisfação com a empresa atual?'].apply(lambda x: 1 if 'Salário' in x else 0)
```

Na coluna de *nível de ensino:*

```
dados['NIVEL DE ENSINO'] = dados['NIVEL DE ENSINO'].apply(lambda x: 0 if x== 'Não tenho graduação formal' else
                                1 if x== 'Estudante de Graduação' else
                                2 if x== 'Graduação/Bacharelado' else
                                3 if x== 'Pós-graduação' else
                                4 if x== 'Mestrado' else
                                5 if x== 'Doutorado ou Phd' else -1)
```

Parte 3:

Vamos fazer a seleção das colunas que utilizaremos em nosso modelo e atribuir:

```
dados = dados[['IDADE', 'GENERO', 'NAO_BRANCA', 'TEMPO_EXPERIENCIA', 'INSATISFACAO', 'SETOR', 'REGIAO ONDE MORA',
       'NIVEL DE ENSINO', 'NUMERO DE FUNCIONARIOS', 'SALARIO','NOVO_NIVEL']]
```

Vamos aplicar o get_dummies:

```
dados = pd.get_dummies(dados, columns=['GENERO', 'SETOR', 'NOVO_NIVEL', 'REGIAO ONDE MORA'], drop_first=True)
```

Vamos separar nosso conjunto de dados:

```
X = dados.drop('SALARIO', axis=1)
y = dados['SALARIO']
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.fit_transform(X_test)
```

Módulo 08 - Para ir além 0/9

COMPLETE E CONTINUE SEU CURSO! →