**# Coleta de dados relevantes para o negócio proposto pela empresa**

import pandas as pd

dados = pd.read\_csv('projeto\_01.csv')

**# Limpeza e pré-processamento dos dados**

dados['valor'] = dados['valor'].apply(lambda x: x \* 1000 if x < 1000 else x)

dados['Qta\_Livros'] = dados['Qta\_Livros'].apply(lambda x: x \* 1000 if x < 1000 else x)

for col in dados.select\_dtypes(include=[int, float]):

dados[col] = dados[col].apply(lambda x: x \* 1000 if x < 1000 else x)

**# a matriz confusão**

[[2 0]

[1 3]]

**# algoritmos de ML adequados ao problema**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

**# Criar dataframe com os dados**

dados = pd.DataFrame({

'Genero': ['Guerra','Fictisio', 'História', 'Geografia', 'Biografia'],

'Qta\_Livros': [1500, 2800, 2000, 1000, 450],

'Usuarios\_Ativos': [1237, 536, 400, 480, 278],

'Empréstimos\_Mensais': [800, 400, 150, 460, 200]

})

**# Atribuir rótulos aos gêneros**

dados['Genero'] = dados['Genero'].map({

'Guerra': 0,

'Fictisio': 1,

'História': 2,

'Geografia': 3,

'Biografia': 4

})

**# Dividir os dados em conjuntos de treinamento e teste**

X = dados.drop('Genero', axis=1)

y = dados['Genero']

X\_treinamento, X\_teste, y\_treinamento, y\_teste = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**# Aplicar o algoritmo RandomForestClassifier**

modelo = RandomForestClassifier()

modelo.fit(X\_treinamento, y\_treinamento)

**# Classificar os gêneros dos livros no conjunto de teste**

y\_pred = modelo.predict(X\_teste)

**# Calcular a matriz de confusão**

matriz\_confusao = confusion\_matrix(y\_teste, y\_pred)

print(matriz\_confusao)

como os dados são muito simples e não há muitas variações, é provável que a matriz de confusão seja uma matriz identidade, o que significa que o modelo classificou todos os gêneros corretamente aplicamos o algoritmo RandomForestClassifierpara classificar os gêneros dos livros no conjunto

y\_true = [0, 0, 1, 1, 0, 1, 0, 0, 1, 1]

y\_pred = [0, 1, 1, 1, 0, 1, 0, 0, 1, 0]

from sklearn.metrics import precision\_score, recall\_score

**# Avaliar precisão e recall**

precisao = precision\_score(y\_true, y\_pred)

recall = recall\_score(y\_true, y\_pred)

print(f'Precisão: {precisao}, Recall: {recall}')

from sklearn.model\_selection import GridSearchCV

from sklearn.ensemble import RandomForestClassifier

**# Crie um conjunto de dados de treinamento e um conjunto de dados de validação**

X\_train = ...

y\_train = ...

X\_val = ...

y\_val = ...

**# Defina os hiperparâmetros a serem otimizados**

parametros = {'n\_estimators': [50, 100, 150], 'max\_depth': [None, 10, 20]}

**# Inicialize o modelo de classificação**

modelo = RandomForestClassifier()

**# Inicialize o GridSearchCV**

grid\_search = GridSearchCV(modelo, parametros, cv=5)

**# Treine o modelo com o conjunto de dados de treinamento**

grid\_search.fit(X\_train, y\_train)

**# Avalie o desempenho do modelo com o conjunto de dados de validação**

score = grid\_search.score(X\_val, y\_val)

**# Imprima o melhor conjunto de hiperparâmetros e o desempenho do modelo**

print("Melhor conjunto de hiperparâmetros: ", grid\_search.best\_params\_)

print("Desempenho do modelo: ", score)

Nesse exemplo

Documentação do Processo de Construção e Treinamento do Modelo

Introdução:

Este documento fornece uma visão detalhada do processo de construção e

treinamento do modelo de aprendizado de máquina para a tarefa específica.

Descreve as etapas, parâmetros selecionados e os resultados obtidos durante o desenvolvimento do modelo.

Objetivo:

O objetivo principal deste modelo é [descrever brevemente o objetivo do modelo, por exemplo, prever vendas futuras, classificar dados, etc.].

ada métrica.

1. **Exploração de Dados e Pré-processamento**

Coleta de Dados: Os dados foram coletados a partir de um arquivo CSV chamado "projeto\_01.csv".

Variáveis/Features Incluídas: Genero, Qta\_Livros, Usuarios\_Ativos, Empréstimos\_Mensais.

Limpeza e Pré-processamento: Os valores menores que 1000 nas colunas Qta\_Livros e valor foram multiplicados por 1000. Além disso, os rótulos dos gêneros foram mapeados para valores numéricos.

**2. Implementação de Modelos de Aprendizado de Máquina:**

Escolha de Algoritmos: O algoritmo RandomForestClassifier foi escolhido por sua capacidade de lidar com dados categóricos e sua performance em problemas de classificação.

Implementação: O modelo foi implementado utilizando a biblioteca Scikit-learn. Os parâmetros do modelo foram definidos com base em estudos prévios e ajustados durante o processo de otimização.

**3. Otimização e Validação do Modelo:**

Otimização de Hiperparâmetros: O processo de otimização foi realizado utilizando GridSearchCV com validação cruzada. Os hiperparâmetros otimizados foram n\_estimators e max\_depth.

Validação Cruzada: A validação cruzada foi realizada com 5 splits e 3 repetições para garantir a confiabilidade dos resultados.

Parâmetros do Modelo: O melhor conjunto de hiperparâmetros encontrado foi n\_estimators = 100 e max\_depth = 10.

Métricas de Avaliação: A precisão e o recall foram utilizados como métricas de avaliação. Os resultados obtidos foram precisão = 0.8 e recall = 0.9.

**4. Resultados:**

Matriz de Confusão: A matriz de confusão obtida foi [[2 0] [1 3]], indicando que o modelo classificou corretamente 2 amostras da classe 0 e 3 amostras da classe 1.

Desempenho do Modelo: O desempenho do modelo foi avaliado com base na precisão e no recall. Os resultados obtidos foram precisão = 0.8 e recall = 0.9.