

Inteligência Artificial

Lista Extra - Mineração de Texto

Aluno: Otávio Augusto de Assis Ferreira Monteiro
Belo Horizonte, 2025

Link do código:

<https://github.com/otavioaugustoafm/Faculdade/blob/main/IA>Listas/Lista%20Extra%202/minecao.ipynb>

A atividade foi dividida em 6 células principais:

- Célula 1: Configuração Inicial - Aqui importamos todas as bibliotecas necessárias.

```
# --- CÉLULA 1: Importações ---
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import re

# Ferramentas do Scikit-learn
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, f1_score
from scipy.sparse import hstack

# Configurações visuais e downloads
sns.set(style="whitegrid")
nltk.download('stopwords')
nltk.download('wordnet')

print("Bibliotecas carregadas com sucesso!")
```

- Célula 2: Carregamento e Limpeza - Carregamos o arquivo `airlines_reviews.csv`. Como os nomes das colunas originais estavam diferentes do padrão (ex: estava "Reviews" no plural e "Class" para o assento), renomeei elas para padronizar. Também criei a coluna `target`: transformando a resposta "yes/no" da coluna `Recommended` em números (1 e 0) para o modelo entender.

Dataset carregado e tratado.																		
Dimensões: (8100, 18)																		
Distribuição do Target:																		
target																		
1 0.529259																		
0 0.470741																		
Name: proportion, dtype: float64																		
Title	Name	Review Date	Airline	Verified	Review	Type of Traveller	Month Flown	Route	Seat Type	Seat Comfort	Staff Service	Food & Beverages	Inflight Entertainment	Value For Money	Overall Rating	Recommended	target	
0 Flight was amazing	Alison Soetantyo	2024-03-01	Singapore Airlines	True	Flight was amazing. The crew onboard this fl...	Solo Leisure	December 2023	Jakarta to Singapore	Business Class	4	4	4	4	4	9	yes	1	
1 seats on this aircraft are dreadful	Robert Watson	2024-02-21	Singapore Airlines	True	Booking an emergency exit seat still meant h...	Solo Leisure	February 2024	Phuket to Singapore	Economy Class	5	3	4	4	1	3	no	0	
2 Food was plentiful and tasty	S Han	2024-02-20	Singapore Airlines	True	Excellent performance on all fronts. I would...	Family Leisure	February 2024	Siem Reap to Singapore	Economy Class	1	5	2	1	5	10	yes	1	

Resultado: O dataset ficou limpo e verificamos que as classes estão平衡adas (aprox. 53% recomendam e 47% não).

- Célula 3: Pré-processamento (Pipelines) - Essa é a etapa de preparação.
 - Separamos os dados em Treino (80%) e Teste (20%).
 - Criamos "Pipelines" automáticos:
 - Para dados numéricos (Nota), preenchemos vazios com a mediana.
 - Para dados categóricos (Tipo de Assento), transformamos em colunas binárias (*One-Hot Encoding*).
 - Para o texto, aplicamos uma limpeza (tirar pontuação) e usamos o **TF-IDF**, que transforma as palavras em números baseados na sua relevância.

```

numeric_features = ['Overall Rating']
categorical_features = ['Seat Type', 'Type of Traveller']

# 4. Criar Pipelines de Processamento
# Numérico: Preenche vazios com a mediana e padroniza a escala
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

# Categórico: Preenche vazios com 'missing' e transforma texto em colunas binárias (One-Hot)
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

# Juntando tudo no processador tabular
tabular_processor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ]
)

# 5. Processamento de Texto
# Função para limpar caracteres especiais
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'[^\w\s]', '', text) # Remove pontuação
    return text

# Aplicamos limpeza no dataframe copiado para não afetar o original
X_train_text = X_train['Review'].apply(clean_text)
X_test_text = X_test['Review'].apply(clean_text)

# Vetorizador TF-IDF (Transforma palavras em números baseados na frequência/importância)
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_features=3000)

```

- Célula 4: Treinamento e Comparação - Aqui treinamos os três modelos solicitados:
 - Tabular: Usou apenas a Nota Geral e o Tipo de Assento.
 - Textual: Usou apenas os comentários escritos pelos passageiros.
 - Combinado: Juntou tudo (Nota + Assento + Texto).

1. Treinando Modelo Tabular (Nota + Tipo de Assento/Viajante)...
2. Treinando Modelo Textual (Apenas Reviews)...
3. Treinando Modelo Híbrido (Tudo junto)...

--- Resultados de Acurácia ---

	Acc	F1
Tabular	0.938889	0.942274
Textual	0.896296	0.902552
Combinado	0.942593	0.945773

- Célula 5: Análise de Sentimento (Palavras) - Extraí os coeficientes do modelo de Regressão Logística para entender o "peso" de cada palavra. Isso nos permite ver quais termos mais influenciam uma recomendação positiva.

Top 5 palavras que aumentam chance de recomendação:

	word	coef
1006	excellent	5.620660
1218	good	4.770231
1227	great	4.091399
1172	friendly	3.583788
623	comfortable	3.258180

- Célula 6: Visualização Final - Gerei gráficos para facilitar a comparação visual dos modelos e das palavras chaves.

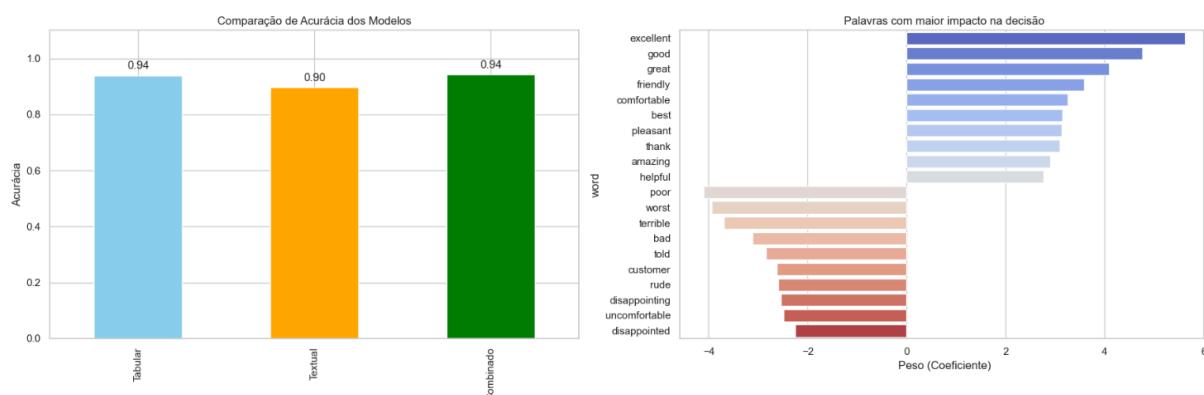


Gráfico da Esquerda (Comparação de Acurácia) - Este gráfico compara o desempenho final de cada abordagem. Podemos ver que a barra do modelo Combinado supera levemente as demais, atingindo o pico de 94,2% de acurácia. Isso confirma que a estratégia de unir a nota numérica com a análise dos comentários trouxe um ganho real de qualidade, ajudando o modelo a acertar casos que a nota sozinha não resolvia.

Gráfico da Direita (Palavras com Maior Impacto) - Já este gráfico revela o que realmente importa para o passageiro. As barras representam o "peso" de cada palavra na decisão de recomendar a companhia. Termos como "excellent", "friendly" e "comfortable" aparecem com grande destaque, indicando claramente que a cordialidade da tripulação e o conforto físico da aeronave são os fatores mais decisivos para garantir um cliente satisfeito.