ANÁLISE DE SENTIMENTO

TWITTER

START

QUAL A PROPOSTA?

01

O objetivo principal deste projeto é desenvolver um sistema de análise de sentimentos que seja capaz de classificar textos do Twitter em quatro categorias principais: irrelevante, neutro, positivo e negativo.

02

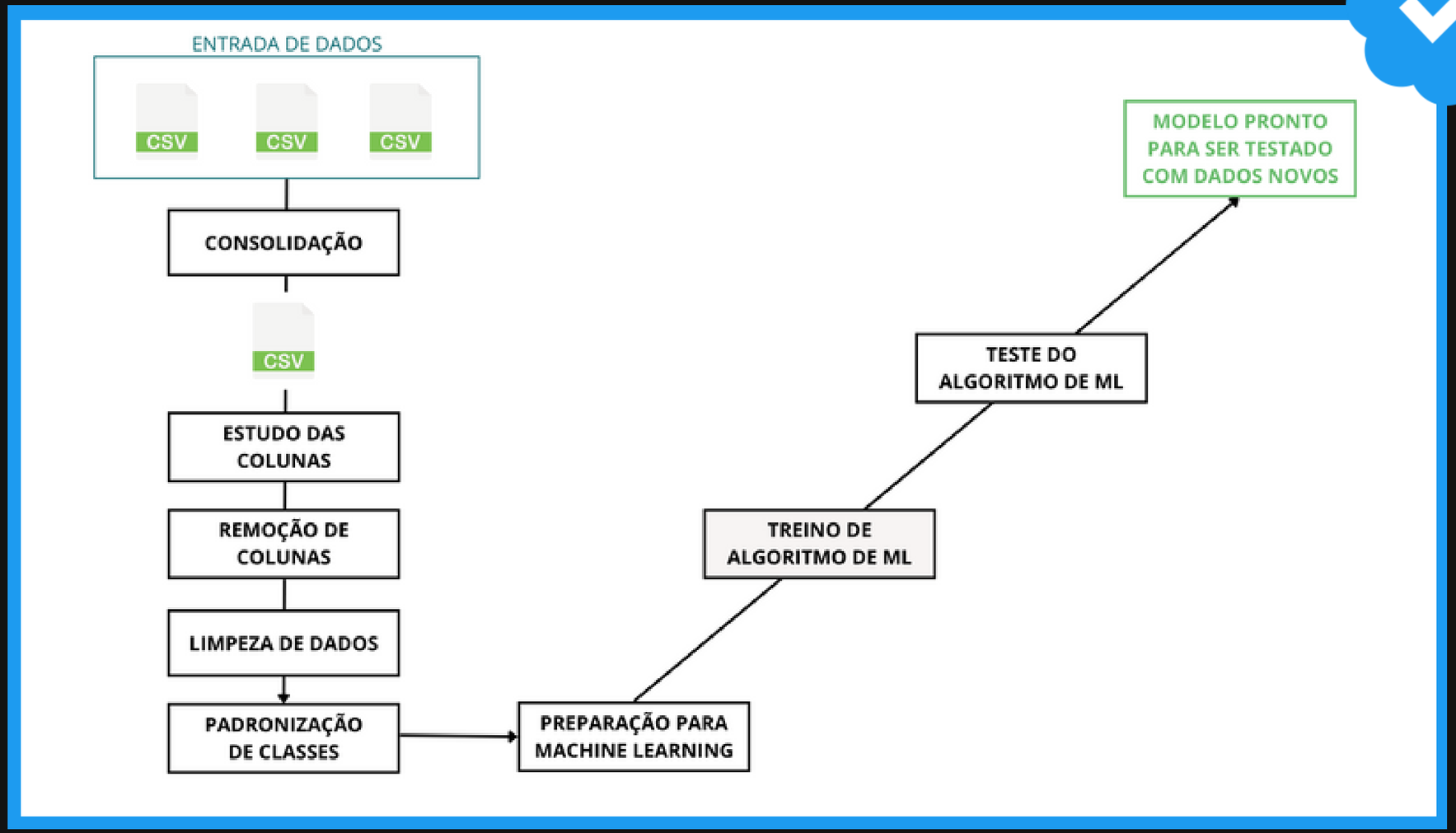
Tomando como fonte de dados Tweets catalogádos em um database CSV, fizemos uma análise de sentimento em cima dos textos.

QUAL A PROPOSTA?

03 A análise visava demonstrar se os **tweets tinham cunho positivo ou negativo** com base nas palavras utilizadas pelo interlocutor.

04 Isso permitirá uma avaliação abrangente do sentimento geral associado a tópicos específicos, marcas, eventos ou tendências no Twitter.

ARQUITETURA

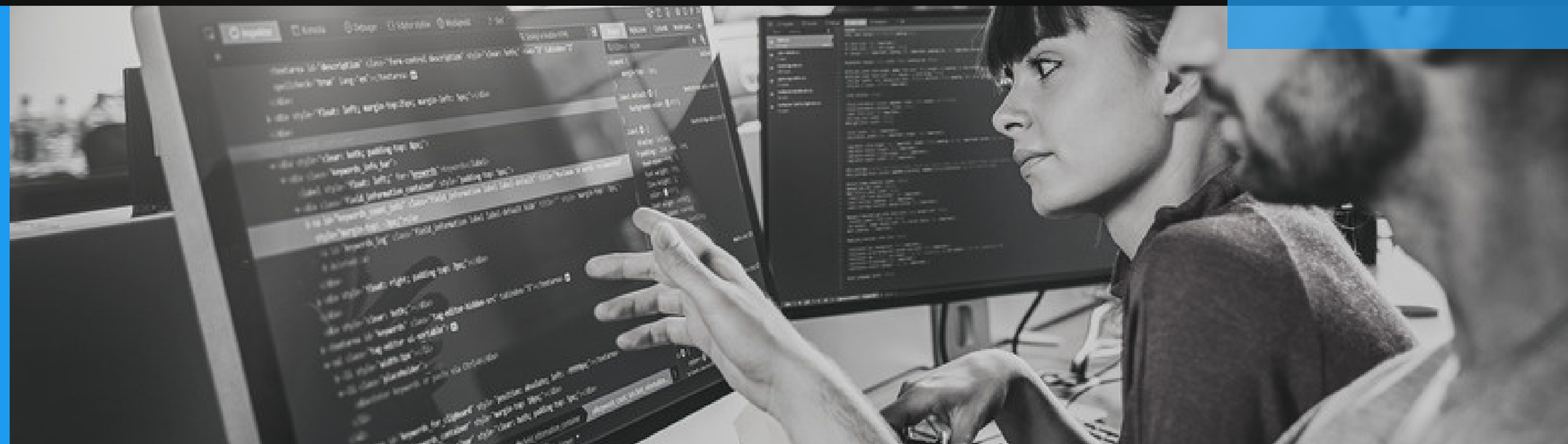


ORIGEM E TRATAMENTO

DOS DADOS

Extraímos três bases de dados do Kaggle, e realizamos as limpezas e normalizações necessárias para a análise.

- **Twitter_Data**
- **Twiter_Training**
- **Train**



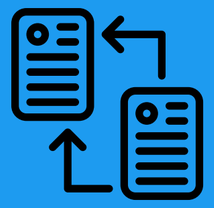
HANDS ON

CÓDIGO NA TELA

TÉCNICAS DE PROCESSAMENTO APLICADAS

Consolidação de arquivos

```
#Concatenar os 3 datasets em um dataframe
df = pd.concat([dataset01,dataset02,dataset03], ignore_index=True)
```



Padronização de classes

```
#Padronizar classes
df.replace(-1,"Negativo",inplace=True)
df.replace(0,"Neutro",inplace=True)
df.replace(1,"Positivo",inplace=True)
df.replace("Negative","Negativo",inplace=True)
df.replace("Neutral","Neutro",inplace=True)
df.replace("Positive","Positivo",inplace=True)
df.replace("negative","Negativo",inplace=True)
df.replace("neutral","Neutro",inplace=True)
df.replace("positive","Positivo",inplace=True)
```

```
In [193…
#Verificar classes
df.Sentiment.unique()

Out[193…
array([-1.0, 0.0, 1.0, nan, 'Positive', 'Neutral', 'Negative',
       'Irrelevant', 'neutral', 'negative', 'positive'], dtype=object)
```

```
In [195…
#Verificar classes
df.Sentiment.unique()

Out[195…
array(['Negativo', 'Neutro', 'Positivo', nan, 'Irrelevant'], dtype=object)
```

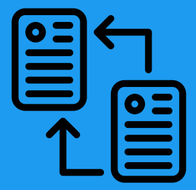
Conversão de Tipo

```
df["Tweet"] = df["Tweet"].astype("str")
df["Sentiment"] = df["Sentiment"].astype("category")
```

TÉCNICAS DE PROCESSAMENTO APLICADAS

Transformaçao de Texto

```
#Transformar texto em letras minúsculas
df["Tweet"] = df["Tweet"].str.lower()
```



Remoção de tweets nulos, duplicados Pontuação e Stopwords

```
In [196…
#Remover NaN e Irrelevant
df.dropna(inplace=True)
df.drop(df[df["Sentiment"]=="Irrelevant"].index,inplace=True)
df.drop(df[df["Sentiment"]=="Neutro"].index,inplace=True)
```

```
#Remover tweets duplicados
df.drop_duplicates(subset=["Tweet"], inplace=True)
```

TÉCNICAS DE PROCESSAMENTO APLICADAS

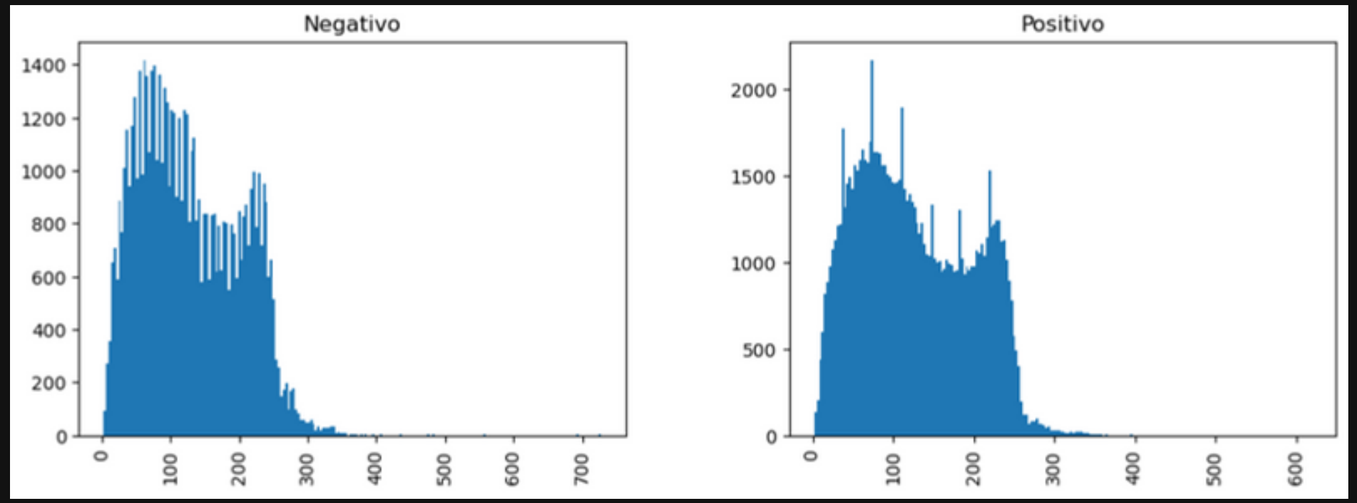
Normalização de classes

```
#Descrição das classes
df.groupby("Sentiment").describe()
```

```
#Verificar tamanho dos tweets
df["Tamanho"] = df["Tweet"].apply(len)
```

```
#Remover os tweets de tamanho 1
df.drop(df[df["Tamanho"] == 1].index,inplace=True)
```

```
#Gráfico por classe
df.hist(column="Tamanho", by="Sentiment", bins=200, figsize=(12,4))
```



Vetorização TF - IDF

Fizemos a vetorização dos tweets, que estavam como listas de tokens (lemmas).

- **CountVectorizer do SciKit Learn**
- **A normalização foi feita com TF-IDF, utilizando *scikit-learn TfidfTransformer***

term frequency-inverse document frequency

TÉCNICAS DE PROCESSAMENTO APLICADAS

TREINO/ TESTE

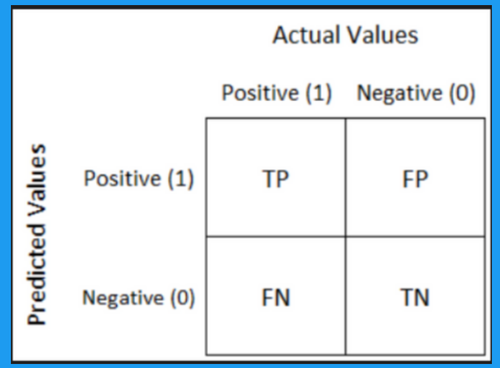
```
msg_train, msg_test, label_train, label_test = train_test_split(df["Tweet"], df["Sentiment"], test_size=0.3, random_state
```

```
print(len(msg_train), len(msg_test), len(msg_train) + len(msg_test))
```

PIPELINE

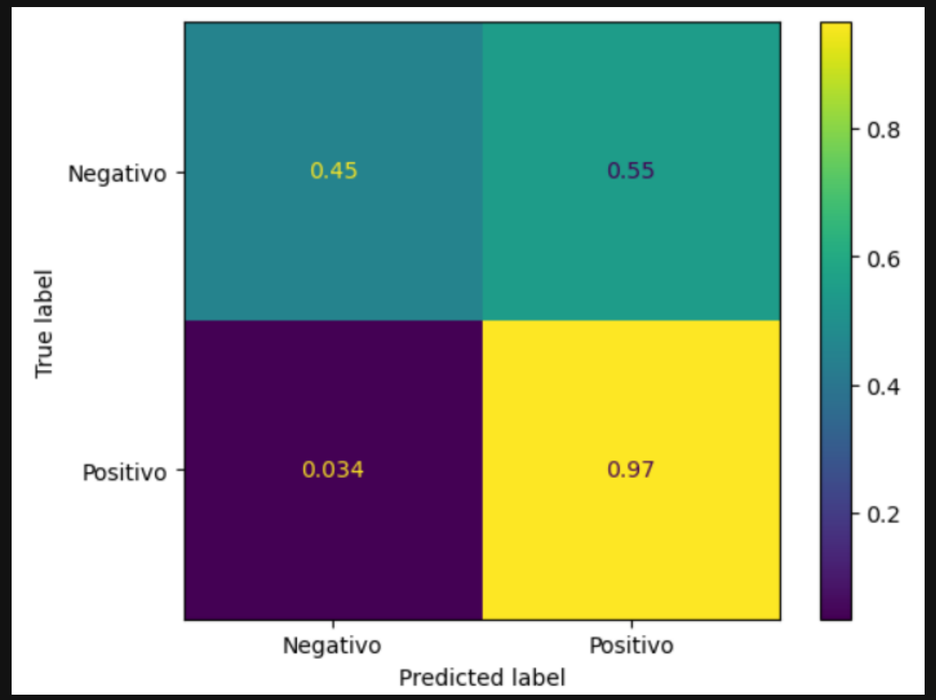
```
pipeline = Pipeline([
    ("bow", CountVectorizer(analyzer=text_process)),
    ("tfidf", TfidfTransformer()),
    ("classifier", MultinomialNB())
])
```

```
pipeline.fit(msg_train, label_train)
```




```
# plotar a confusion matriz
mc = ConfusionMatrixDisplay(
    confusion_matrix=confusion_matrix(label_test, predictions, normalize="true"), display_labels=pipeline.classes_)
mc.plot()
plt.show()
```

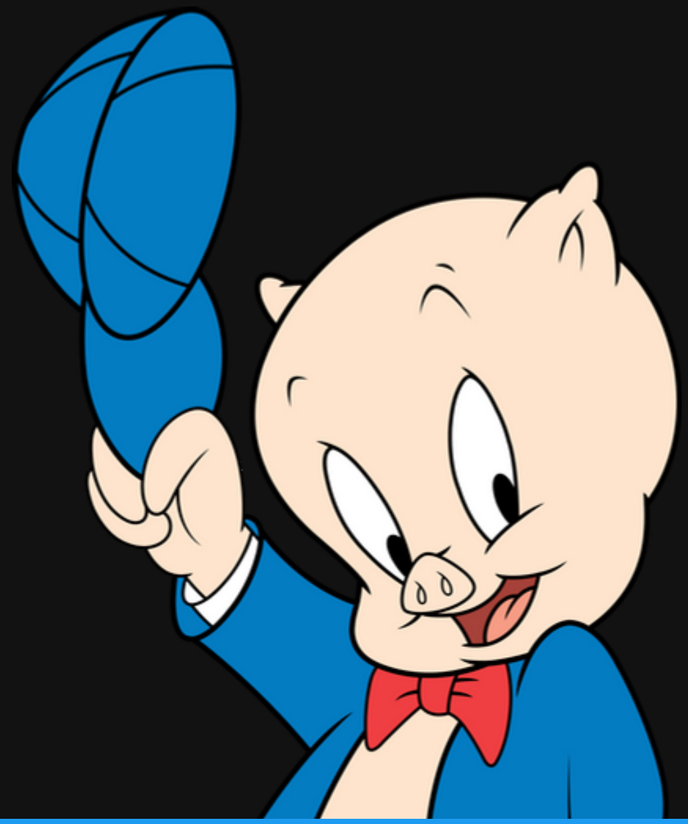
ANÁLISE DOS RESULTADOS



	precision	recall	f1-score	support
Negativo	0.45	0.90	0.60	9768
Positivo	0.97	0.73	0.83	39483
accuracy			0.76	49251
macro avg	0.71	0.81	0.72	49251
weighted avg	0.86	0.76	0.79	49251

CONCLUSÃO

A detecção de sentimentos negativos em tweets possibilita a identificação precoce de sinais de saúde mental, facilitando intervenções e apoio apropriado. A análise do sentimento em torno de questões sociais e políticas oferece insights preciosos sobre o engajamento cívico e as perspectivas da juventude em relação a temas relevantes.



THAT'S ALL FOLKS