

---

# PROJETO 2

---

CLASSIFICADOR NAIVE-BAYES

---

## CLASSIFICADOR AUTOMÁTICO DE SENTIMENTO

Você foi contratado por uma empresa para analisar como os clientes estão reagindo a um determinado produto no Twitter. A empresa deseja que você crie um programa que selecione algumas mensagens disponíveis no Twitter que mencionam esse produto e classifique-as como "relevante" ou "irrelevante".

Com isso, essa empresa deseja que mensagens relevantes, que denigrem o nome do produto, ou que mereçam destaque, disparem um foco de atenção da área de marketing.

Como aluno de Ciência dos Dados, você lembrou do Teorema de Bayes, mais especificamente do Classificador Naive-Bayes, que é largamente utilizado em filtros anti-spam de e-mails, por exemplo. Esse classificador permite calcular qual a probabilidade de uma mensagem ser relevante dada as palavras em seu conteúdo.

Para realizar o MVP (*minimum viable product*) do projeto, você precisa implementar uma versão do classificador que "aprende" o que é relevante com uma base de treinamento e compara a performance dos resultados com uma base de testes.

Após validado, o seu protótipo poderá também capturar e classificar automaticamente as mensagens da plataforma.

### Preparando o ambiente no Jupyter:

Instale a biblioteca *tweepy* para realizar a conexão com o Twitter:

```
%capture
#Instalando o tweepy
!pip install tweepy
```

Importe as bibliotecas que serão utilizadas neste projeto. Esteja livre para adicionar outras bibliotecas que julgar necessárias:

```
import tweepy
import math
import os.path
import pandas as pd
import json
from random import shuffle
```

## **Autenticando o Twitter:**

Para realizar a captura dos dados no Twitter, é necessário ter uma conta cadastrada no twitter como **Student**:

1. Siga os passos direcionados no arquivo Guia.pdf para criar uma conta e registrar um app.
2. Finalizado esse guia, dentro do registro do App, na aba Keys and Access Tokens, anotar os seguintes campos:
  - ✓ Consumer Key (API Key)
  - ✓ Consumer Secret (API Secret)
3. Mais abaixo, gere um Token e anote também:
  - ✓ Access Token
  - ✓ Access Token Secret
4. Preencha os valores no arquivo "auth.pass"

**ATENÇÃO:** Nunca divulgue os dados desse arquivo online (GitHub, etc). Ele contém as chaves necessárias para realizar as operações no twitter de forma automática e portanto é equivalente a ser "hackeado". De posse desses dados, pessoas mal intencionadas podem fazer todas as operações manuais (tweetar, seguir, bloquear/desbloquear, listar os seguidores, etc). Para efeito do projeto, esse arquivo não precisa ser entregue!!!

## **Etapas do projeto:**

Após ter a conta criada no twitter, deve seguir os seguintes passos:

### **1. Escolha de um produto e coleta das mensagens**

Após escolher um produto, vamos coletar os dados: *tweets*. Tenha em mente que dependendo desse produto escolhido, não haverá uma quantidade significativa de mensagens, ou ainda poder haver muitos *retweets*.

**Commented [KV1]:** Possivelmente aqui entrará as fotos que Fabio fez para esse roteiro.

---

As mensagens (500 no total) serão salvas em uma planilha no Excel.

## 2. Classificando as mensagens na coragem:

Agora você deve abrir o arquivo Excel com as mensagens capturadas e classificar cada mensagem como **relevante (valor 1) ou irrelevante (valor 0)**. Guarde essa classificação na coluna **B**, colocando um nome para essa coluna na célula **B1**.

Fazer o mesmo na planilha de Teste.

## 3. Montando o classificador Naive-Bayes:

Considerando apenas as mensagens da planilha Treinamento, o objetivo aqui é ensinar o seu classificador quais são as palavras mais comuns (frequentes) em uma mensagem **relevante** e as mais presentes nas mensagens **irrelevantes**. Nesse caso, seu código deve conter:

- ✓ Limpeza de mensagens removendo os caracteres: enter, :, ", ', (, ), etc. Não remover emojis.
- ✓ Correção de espaços entre palavras e/ou emojis.
- ✓ Proposta

## 4. Verificando a *performance*:

Considerando agora apenas as mensagens da planilha Teste, seu objetivo aqui é testar a qualidade do seu classificador.

Para tanto, você deve extrair as seguintes contagens:

- ✓ Porcentagem de verdadeiros positivos (mensagens relevantes e que são classificadas como relevantes)
- ✓ Porcentagem de falsos positivos (mensagens irrelevantes e que são classificadas como relevantes)
- ✓ Porcentagem de verdadeiros negativos (mensagens irrelevantes e que são classificadas como irrelevantes)
- ✓ Porcentagem de falsos negativos (mensagens relevantes e que são classificadas como irrelevantes)

Opcionalmente:

- ✓ Criar categorias intermediárias de relevância baseado na diferença de probabilidades. Exemplo: muito relevante, relevante, neutro, irrelevante e muito irrelevante.

## 5. Concluindo:

---

Faça um comparativo qualitativo sobre os percentuais obtidos para que possa discutir a *performance* do seu classificador.

Explique como são tratadas as mensagens com dupla negação e sarcasmo.

Proponha um plano de expansão. Por que eles devem continuar financiando o seu projeto?

Opcionalmente:

- ✓ Discorrer por que não posso alimentar minha base de Treinamento automaticamente usando o próprio classificador, aplicado a novos tweets.
- ✓ Propor diferentes cenários de uso para o classificador Naive-Bayes. Pense em outros cenários sem intersecção com este projeto.
- ✓ Sugerir e explicar melhorias reais no classificador com indicações concretas de como implementar (não é preciso codificar, mas indicar como fazer. Indique material de pesquisa sobre o assunto).

## **REGRAS:**

1. O Projeto 2 é idealmente **em dupla**. Trios podem ser considerados, mas com rubrica mais rigorosa.
2. Use o **arquivo *layout*** disponibilizado na pasta Projeto2 do Github e Blackboard.
3. Os entregáveis via GitHub:
  - ✓ Arquivo notebook com o código do classificador, seguindo as orientações do *layout*.
  - ✓ Arquivo Excel com as mensagens de treinamento e teste totalmente classificadas.

**\*\*NÃO disponibilizar o arquivo com os *\*access keys/tokens\** do Twitter.\*\***

**A estrutura do documento deve ser clara e de fácil compreensão da linha de raciocínio. Nesse caso, o notebook não deve haver excesso de impressões não discutidas de variáveis e de *dataframe*.**

**Aconselhamos fazer uma análise geral e, após finalizada, salve com outro nome, limpe seu IPython Notebook apenas com os resultados relevantes e melhore seu texto.**

---

## **CRONOGRAMA:**

### **Turmas A, B e C**

<b>DATA</b>	<b>Finalização:</b>
<b>10/09</b>	APS2: deve estar no Github até às 23h59 com as seguintes evidências: <ul style="list-style-type: none"><li>✓ Conta no twitter criada.</li><li>✓ Produto escolhido.</li><li>✓ Arquivo Excel contendo a base de treinamento e teste já classificado.</li></ul>
<b>12/09</b>	Os entregáveis via GitHub: <ul style="list-style-type: none"><li>✓ Arquivo notebook com o código do classificador, seguindo as orientações do <i>layout</i>.</li><li>✓ Arquivo Excel com as mensagens de treinamento e teste totalmente classificadas.</li></ul>