PROJETO 1

CLASSIFICADOR NAIVE-BAYES COM TWEETS

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, as quais mencionam esse particular produto; e classifique esses *tweets* como "relevante" ou "irrelevante", pelo menos.

Com isso, essa empresa deseja que mensagens relevantes, que denigrem o nome do produto, ou que mereçam destaque, por exemplo, 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 poderia, porque não, também capturar e classificar automaticamente as mensagens da plataforma.

Preparando o ambiente no jupyter:

Pelo arquivo **Projeto1_Obtenção_dos_tweets.ipynb**, 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, anotar os seguintes campos:
   - ✓ Consumer Key (API Key)
   - ✓ Consumer Secret (API Secret)
   - ✓ Access Token
   - ✓ Access Token Secret
3. Preencha os valores no arquivo "auth.pass"

ATENÇÃO: Nunca divulgue os dados desse arquivo (auth.pass) 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*.

   As mensagens (500 no total) serão salvas em uma planilha no Excel. Se você for usar três categorias, use pelo menos 750 no total.

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.

   Um ponto de atenção nesta etapa é evitar que cada membro do grupo classifique os tweets com um critério diferentes. Conversem e garantam que o critério que define as

mensagens como relevante (valor 1) ou irrelevante (valor 0) esteja bem definido entre os membros do grupo.

Caso haja um percentual muito baixo de mensagens relevantes ou de irrelevantes, selecionar mais *tweets* da categoria com baixo percentual. É importante que haja quantidades parecidas de mensagens relevantes e irrelevantes na base de dados treinamento.

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 preferencialmente:

- ✓ Limpeza de mensagens removendo os caracteres: enter, :, '', ', (,), etc. Não remover emojis.
- ✓ Correção de espaços entre palavras e/ou emojis.
- ✓ Proposta de outras limpezas/transformações que não afetem a qualidade da informação.
- ✓ Suavização de Laplace: link1 (com leitura até **antes** da seção "Creating a naive bayes classifier with Monkeylearn") e link2.

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)
- ✓ Acurácia (mensagens corretamente classificadas, independente da categoria)

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).

6. **Qualidade do Classificador a partir de novas separações dos *tweets* entre Treinamento e Teste**

   Um importante passo no aprendizado de máquina é trabalhar com uma boa base de dados para o treinamento e teste do seu classificador. Entretanto, é razoável pensar que a divisão de dados utilizada no seu Classificador representa uma entre muitas possíveis combinações em dividir o total de *tweets* em 300 para treinamento e 200 para teste.

   Assim sendo, aqui o objetivo é avaliar como os *tweets* contidos na base de dados treinamento pode interferir numa melhor ou não tão boa classificação das mensagens contidas na base de teste.

   Nesse caso, faça:

   - ✓ Junte os 500 tweets em único *dataframe* e separe, de forma aleatória, 300 tweets para fica na base de dados treinamento e 200, na base de dados teste. **Obs.: Apenas aqui sua dupla poderá usar alguma biblioteca que possua um comando já pronto que realiza essa separação na base de dados (split em train e test)**;
   - ✓ Para cada base separada, faça os itens de 3 a 4 descritos no tópico **Etapas do projeto** e guarde os percentuais de acertos (= % de positivos verdadeiros + % de negativos verdadeiros);
   - ✓ Repita os dois passos acima 100 vezes.

   Construa um histograma com esses percentuais de acertos e discuta o resultado do histograma refletindo sobre possíveis vantagens ou desvantagens sobre construir um

Classificador considerando uma única vez a divisão da base de dados em treinamento e em teste.

REGRAS:

1. O Projeto 1 é em DUPLA. No caso de TRIO, terá rubrica diferente para seguir.
2. O projeto será corrigido conforme os critérios da rubrica.
3. Use os **notebooks** disponibilizados na pasta Projeto 1 do Github e Blackboard.
4. Os entregáveis via GitHub:
   - ✓ Arquivos notebooks com o código para obter os twets do Twitter e com código do classificador, seguindo layout dos notebooks disponibilizados na pasta Projeto 1.
   - ✓ Arquivo Excel com as mensagens de treinamento e teste totalmente classificadas.

****NÃO disponibilizar o arquivo com os *access keys/tokens* do Twitter, leia sobre .gitignore****

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

DATA	**Finalização:**
04/09 (sábado)	Preencher o google forms até às 23h59 com as seguintes evidências: ✓ Dupla ou trio formado. ✓ Repositório vazio no github criado. Link para entrega: https://forms.gle/YdFmLrzNACA2Dwch7
08/09 (quarta)	Deve estar no Github até às 23h59 com as seguintes evidências: ✓ Conta no twitter criada. ✓ Produto escolhido. ✓ Arquivo Excel contendo a base de treinamento e teste já classificados.
23/09 (quinta)	Os entregáveis via GitHub: ✓ Arquivos notebooks com o código para obter os *tweets* do Twitter ✓ Arquivo Excel com as mensagens de treinamento e teste totalmente classificadas. ✓ Arquivo com o código do classificador e análise dos resultados, seguindo *layout*.

RUBRICA:

Nível	Descrição
I	Não entregou; Entregou, mas não tem sequer a base de tweets para treinamento e teste A base de testes não tem rótulos feitos manualmente
D	Tem a base de tweets para treinamento e testes, mas não funciona Existem rotinas para cálculos de probabilidades, mas as fórmulas ou cálculos estão errados, ou não funciona
C	Entregou; Tem a base de tweets para treinamento e testes; Limpou: \n, :, ", ', (,), etc SEM remover emojis; Rotinas funcionam, mas análise não ficou completa; ou não ficou boa; ou não faz a suavização de Laplace (Smoothing) corretamente
B	Entregou; Tem a base de tweets para treinamento e testes; Limpou: \n, :, ", ', (,), etc SEM remover emojis; Faz a suavização de Laplace (Smoothing) corretamente; Utilizou métricas adequadas para a análise da qualidade do classificador (item **4. Verificando a performance** do enunciado); Produziu um texto de qualidade na análise crítica da performance do classificador (item **5. Concluindo** do enunciado); Mas o projeto carece de melhorias (ver itens avançados)
CASO SEU PROJETO SE ENQUADRE EM ALGUM DOS NÍVEIS ACIMA, ENTÃO OS ITENS AVANÇADOS SERÃO IGNORADOS; **SENÃO, SEU NÍVEL SERÁ PELA CONTAGEM DE ITENS AVANÇADOS:** **B+ : 3 itens** **A : 4 ou 5 itens** **A+ : 6 ou 7 itens**	IMPLEMENTOU outras limpezas e transformações que não afetem a qualidade da informação contida nos tweets. Ex: stemming, lemmatization, stopwords
	CORRIGIU separação de espaços entre palavras e emojis ou entre emojis e emojis
	CRIOU categorias intermediárias de relevância baseadas na probabilidade: ex.: muito relevante, relevante, neutro, irrelevante, muito irrelevante. Pelo menos quatro categorias, com adição de mais tweets na base, conforme enunciado. (**OBRIGATÓRIO PARA TRIOS, sem contar como item avançado)**
	EXPLICOU porquê não pode usar o próprio classificador para gerar mais amostras de treinamento
	PROPÔS diferentes cenários para Naïve Bayes fora do contexto do projeto
	SUGERIU e EXPLICOU melhorias reais com indicações concretas de como implementar (indicar como fazer e indicar material de pesquisa)
	FEZ o item **6. Qualidade do Classificador a partir de novas separações dos tweets entre Treinamento e Teste** descrito no enunciado do projeto (OBRIGATÓRIO para conceitos A ou A+)